Research Article

MFRR-CEF: Overlap- and Entropy-Aware Feature Ranking With Calibrated Ensemble Fusion For Imbalanced Multiclass Iot Intrusion Detection

Sandhya Sukhabogi[1] , Dr. M. Anusha[2]

[1]Research Scholar , Department of CSE,Koneru Lakshmaiah Education Foundation, Vaddeswarm, Guntur District, A.P., INDIA.
[2]Associate Professor, Department of CSE, Koneru Lakshmaiah Education Foundation, Vaddeswarm, Guntur District, A.P., INDIA.

Abstract

Internet of Things intrusion detection involves accurate identification of rare attack classes in the presence of skewed traffic distributions, redundant flow attributes, overlapping behaviors, noisy measurements, and changing class priors. A classifier with high aggregate accuracy may miss theft, ransomware, man-in-the-middle, or backdoor traffic if those classes constitute a small proportion of the observations. This paper introduces MFRR-CEF, an overlap and entropy-aware feature ranking framework with calibrated ensemble fusion for multiclass intrusion detection on the NF-BoT-IoT and NF-ToN-IoT NetFlow benchmarks. The processing pipeline consists of six interconnected steps: leakage-safe flow preparation, robust outlier control, Multi-Class Feature Rank Ratio Plus (MFRR+) scoring, boundary-guarded class balancing, an entropy-weighted novel support vector classifier (NSVC), an enhanced Rotation Forest (ERF), and validation-calibrated diversity fusion. MFRR+ integrates pairwise standardized class separation, density overlap, class rarity, label entropy, and correlation redundancy into a single bounded score. The balancing phase only allows a synthetic minority sample if the local class geometry and a margin test support its label. NSVC assigns class-sensitive costs to one-versus-one margins, and ERF creates supervised feature groups whose retained principal directions are geared towards between-class variation. Fusion weights are derived from inner-fold macro F1, calibration loss, and pairwise error diversity. Six algorithms and corresponding propositions define bounded ranking, preservation of neighborhoods of accepted samples, minority-sensitive margin allocation, orthonormal rotation, convex fusion risk, and leakage isolation. Public class distribution indicates severe skew: NF-BoT-IoT has 470,655 reconnaissance flows but only 1,909 theft flows and NF-ToN-IoT has 468,539 injection flows but only 142 ransomware flows. Compared to the best reported baseline, MFRR-CEF obtains the comparative experiment accuracy of 0.989 and precision of 0.990, while the best reported baseline gets the precision of 0.985 and the experiment accuracy of 0.980. A distinct 699-instance pilot record is an improvement from 674 to 687 correct decisions, which translate to accuracies of 0.9642 and 0.9828 as well as Wilson intervals of [0.9477, 0.9757] and [0.9702, 0.9902]. Therefore, fold-level predictions are not available and comparative significance is not claimed. The architecture, pilot interpretation, error controls, and nested evaluation scheme together give a reproducible framework for a full multiclass benchmark.

Index Terms— class imbalance, ensemble learning, feature ranking, Internet of Things, intrusion detection, NetFlow, NF-BoT-IoT, NF-ToN-IoT, support vector classifier.

I. INTRODUCTION

Sensors, controllers, appliances, gateways, vehicles, cameras, and industrial endpoints connected to the Internet generate continuous streams of operational data. Their installation further increases the range of available services, firmware stacks, communication protocols, and trust relations. IoT endpoints typically run in constrained memory environments, have limited update support, are provided by various vendors, and have lengthy replacement cycles. A compromised IoT device can be used to perform other malicious activities such as participating in denial-of-service attacks, attempting to guess credentials, scanning, stealing data, moving laterally, communicating with command-and-control servers, or even potentially manipulating physical systems. Network layer intrusion detection continues to be useful in this scenario since a flow monitor can monitor communications without needing a large agent on every end point. Promising results are reported by most recent works on IoT security employing machine learning, deep learning, ensemble classification, and distributed detection, yet they also report sensitivity to selected features, traffic heterogeneity, class skew, and resource limits [1], [2], [3], [4].

The quality of the evaluation is contingent upon the traffic evidence that is presented to the detector. Survey of datasets reveals that A/N network intrusion corpora are diverse w.r.t. topology, duration of collection, feature set, taxonomy of attacks, way of labeling, protocol mix and benign traffic [5]. A model can learn to recognize experiment-unique addresses, ports, capture schedules, duplicate records, or artifacts of generator rather than the attack behavior. Sommer and Paxson stated the closed-world problem in ML intrusion detection: a laboratory test

can bias training and test traffic to be over-similar, and can mask operation uncertainty [6]. A believable detector requires leakage-resistant splitting, training-fold feature engineering, per-class metrics, calibration checks, and an explicit consideration of evidence that is missing.

The BoT-IoT testbed was built at UNSW Canberra and includes normal traffic mixed with DDoS, DoS, reconnaissance, keylogging, and data-exfiltration traffic [7]. The source packet captures are over 72 million records, and the five-percent subset that is commonly used has three million records. The ToN-IoT program delivers heterogeneous network, operating system, telemetry, and IoT/IIoT sources from a Cyber Range and IoT testbed [8], [9], [10]. NetFlow conversions enable common header-derived features to be used across different corpora [11]. The standard feature paper by Sarhan, Layeghy, and Portmann shows why a common flow schema helps comparison between datasets and that multiclass recognition can be more difficult than binary classification [12]. The NF-BoT-IoT and NF-ToN-IoT variants used in this work are equipped with attack labels in a unified NetFlow representation.

Class distributions form two distinct imbalance regimes. The public first-version distribution of NF-BoT-IoT contains 600,100 flows: 470,655 reconnaissance, 56,844 DDoS, 56,833 DoS, 13,859 benign, and 1,909 theft flows. The ratio of reconnaissance to theft flows is roughly 246.5:1. NF-ToN-IoT has 1,379,274 flows, distributed among benign and nine attack classes. 468,539 flows are injection, while 142 are ransomware. The ratio of these two classes is about 3,299.6:1. An empirical risk objective, commonly used, can reduce its average loss by fitting to high-frequency classes while accepting high loss on classes that contribute few terms. Micro accuracy can be high while macro recall is low and minority-class precision is unusable. This problem is well studied in the imbalanced-learning literature [13], [14], but the multiclass IoT traffic adds significant class overlap and feature correlation.

Class imbalance is not a single numerical condition. The minority class may be compact and well separated, sparse and fragmented, represented by mislabeled and duplicated flows, or embedded inside a majority region. Random oversampling reproduces the same samples. Random undersampling throws away benign or common-attack structure. Unconstrained interpolation can introduce a synthetic flow between minority instances that are on opposite sides of another class. Cost weighting can increase minority influence, but extreme inverse-frequency weights can cause optimization to become unstable. The choice of a balancing technique depends on local geometry, class support, and the metric used for final comparison.

Feature selection also depends on the class geometry. A global correlation with the label may be a feature that separates benign traffic from the largest class of attack, but does not differentiate rare attacks. Univariate ranking may also select a cluster of highly correlated counters. Standardized mean separation identifies differences in location, but equal means do not mean equal distributions. Mean information captures nonlinear association but can be biased in small cells and is limited in the information it provides about which class pair is confused. We introduce MFRR+, a score that integrates evidence from five sources that are often complementary: pairwise separation, distributional overlap, inverse support, entropy, and redundancy. The pairwise nature of the score ensures that the largest class does not determine the entire ordering.

Outlier treatment is just as important. In the case of intrusion detection an extreme data point could be a corrupt row, a measurement artifact, or the center of evidence of an attack. A global elimination of rare points might delete exactly those events which the model should try to learn. The robust range checks and class-conditional anomaly scores are MFRR-CEF only applied within a training fold. Each flow is deleted only if a rule detects invalid syntax or if its anomaly score surpasses a class-dependent threshold and no protected minority rule is applicable. Continuous features are not standardized using the entire dataset but instead clipped at training set determined limits. Addresses, row IDs, capture-specific sequence fields, and direct label proxies are removed prior to ranking.

The classifiers stage employs two models of different inductive biases. NSVC is a class-sensitive version of the one-versus-one support vector classifier. It applies a larger penalty to a class with smaller effective support, and calibrates pairwise margins prior to multiclass coupling. ERF is a boosted Rotation Forest. It divides selected attributes into supervised clusters, derives class-stratified bootstrap samples, sticks to principal directions in line with a discriminative variance criterion, and constructs balanced decision trees in rotated spaces. NSVC represents smooth maximum-margin boundaries; ERF represents nonlinear partitions and interactions. They do not need to make the same mistakes. A calibrated fusion layer uses validation macro F1, log loss and disagreement to compute nonnegative weights. The final output includes the attack class, calibrated confidence, and a review flag for it low-confidence flows.

The study addresses five research questions.

RQ1: Can pairwise entropy- and overlap-aware ranking select a relatively small number of features without the information ordering being dominated by common classes or by redundant flow attribute?

RQ2: Can boundary-guarded balancing improve rare-class support without placing synthetic flows inside a competing-class neighborhood?

RQ3: Do class-sensitive NSVC margins and supervised Rotation Forest projections offer complementary errors on imbalanced multiclass traffic?

RQ4: Does calibrated diversity fusion improve the reported accuracy and precision comparison relative to individual tree and boosting baselines?

RQ5: What inferences can be supported by the pilot summaries and what further evidence at the fold level is necessary for complete inference validation?

Research questions The above analytical objectives entail seven contributions. The first si a leakage-free flow-based preparation protocol that performs consisting imputation, scaling, outlier clipping, ranking, balancing, calibration and threshold selection on training data. The second is a multi-class ranking measure, named MFRR+, that investigates every class pair and discounts overlap and correlation redundancy. Third, we introduce a boundary-guarded synthetic balancing procedure that discards ambiguous synthesized examples. Fourth, we propose an entropy-weighted NSVC model with a priori information of class costs. The fifth is an ERF with supervised feature groups, class-stratified bootstraps, and Fisher-guided component retention. Sixth, a calibrated fusion objective that favors macro-level validation quality and correlated classifier diversity. Finally, we present an evidence-aligned analysis that separates the four-method comparison from two 699-instance pilot runs and provides a nested statistical procedure for full–data execution in the future.

There are two types of evidence in the experimental record. The first is a plug-in plot with accuracy and precision for the rf, dt, xb, and mfrr-cef. The second is formed by two classifier reports on 699 instances, with details about the number of correct decisions, the value of Cohen's kappa, the mean absolute error, the root mean square error, class-level precision and recall, the Matthews correlation coefficient, the receiver-operating-characteristic area and the precision-recall area. There are no predictions at row-level, no fold indicators, no timing logs, no seeds to randomize from associated with these summaries. The paper reports the actually-observed point estimates, calculates binomial Wilson intervals from the number of correct counts, and makes no paired significance claim. In Section VI, we specify the data and protocol for realizing that claim.

The rest of the paper is organized as follows. II provides an overview of imbalanced learning, feature selection, outlier detection, ensemble classifiers, IoT intrusion detection, calibration, and reporting. Describes the data model, leak boundary, metrics, and hypotheses in Section III. IV introduces the MFRR-CF. Provide five algorithms and mathematical propositions." "IV V gives six algorithms and mathematical propositions. VI describes the experimental, pilot and statistical methodology. VII THE RESULTS The results for are reported. Section VIII addresses minority class behavior in validity, deployment, and failure mode analysis. Presents the reproducibility specification, IX and concludes the study, X.

II. RELATED WORK

A. Imbalanced Multiclass Learning

Unequal error rates-costs and decision region representations are at least part of the issue with imbalanced classification when combined with varying class sizes. Krawczyk also remarks that data distribution, evaluation, online adaptation, and multiclass nature still pose research challenges [15]. Initial research by Japkowicz and Stephen on class skew revealed that class skew is not influenced as an isolated count ratio because it interacts with concept size, sample size, and classifier bias [16]. In network traffic, the unique behavior pattern contained in the small est class may also overlap many common classes in several feature subspaces. A detector has to consider support and overlap.

The data-level approaches alter the training sample distribution. SMOTE generates a synthetic minority sample by taking the difference between a minority instance and one of its nearest minority neighbors [17]. ADASYN generates more synthetic samples in the neighborhood of harder minority samples [18]. Both may contribute useful support, but neither definitively establishes that an interpolated flow is still plausibly or not enveloped by a competing-class neighborhood. RUSBoost (random undersampling boosting) applies random undersampling to the

boosting framework and treats the imbalance problem at the bagging level [19]. Hybrid methods combine sampling, cleaning and cost weighting. Their advantage depends on class fragmentation and contamination.
Ensemble methods have the potential to attenuate variance and to cover distinct regions of an imbalanced border. The survey by Galar et al. categorizes bagging-, boosting-, and hybrid-based ensembles for class imbalance [20]. Random Forest builds decorrelated trees based on bootstrap samples and random subsets of features [21]. Its majority vote is stable in many tabular tasks, yet vanilla bootstrap sampling can only lead to limited representation of a class in the case of several hundred flows. Rotation Forest applies principal-component rotations to attribute subsets before tree induction and it can enhance diversity without throwing away all attributes [22]. The current version of ERF modifies group formation and component selection in a way the rotation is guided by class discrimination and rare-class support.
Support vector machines maximize margin geometry and are known to work well in high dimensional spaces [23]. A typical soft-margin objective is to apply equal cost for all the violations. When supports are uneven, a boundary may be pushed into a rare class at minimal increase in total loss. Weighted SVMs over classes impose different costs, whereas one-versus-one decomposition limited each subproblem to two classes. pairwise models (c) also matches the pairwise construction of MFRR+. Calibration must be performed post margin learning, since an SVM decision value is not a posterior probability.
The multiclass setting creates extra failure modes. One-versus-rest resampling constructs a different imbalance problem for each label and can produce inconsistent posteriors. One-versus-one models create $C(C-1)/2$ pairwise decisions, each with its own support and ambiguity. A global oversampling target may inflate a moderately rare class and leave an extreme class underrepresented. Macro F1 and macro recall assign equal weight to classes at evaluation time, but the training objective and sampling policy must also create meaningful gradients for all classes.

B. Feature Ranking and Redundancy Control

Feature ranking can reduce the number of features and thus the measurement, storage, training, and inference costs, and can help to remove features that have unstable correlations. ReliefF calculates relevance based on feature differences between neighboring instances of same and different classes [24]. It is able to model local interactions, but the quality of the results relies on the choice of distance and neighborhood composition. Minimum Redundancy Maximum Relevance selection seeks to select features that are highly associated with the target while minimizing redundancy among selected features [25]. Guyon and Elisseeff [26] list the goals of variable selection, wrappers, filters, embedded methods and warn that selection should be assessed within the validation.
Intrusion-flow attributes are duration, packet and byte totals, interarrival statistics, TCP flags, protocol values, service ports, and directional counters. Many are mathematically related. Bytes per second can be a function of bytes and duration; packet rates can be a function of packet counts and duration; source and destination attributes can be highly correlated. A ranker that only picks the highest scores may be showing multiple instances of the same thing while missing a lower scoring but complementary attribute. This can be mitigated by using correlation pruning or a soft redundancy penalty.
Overlap between classes is not equivalently is redundancy Two classes could have similar means and different tails or different means and broad overlapping support. A density-overlap coefficient adds up the minimum of two estimated marginal densities and it takes values between zero and one. I can discount standardized separation when there is a wide overlap. Due to noise in density estimates for a small class, the procedure requires a bandwidth control, a fall back method based on quantile intervals and a effective-support limit. MFRR+ derivation employs a smoothed histogram or kernel estimate, which is chosen only from the training data and limits the contribution to the rarity.
Anomalous observations can bias a univariate score; Isolation Forest is a partition-based anomaly measure [27], and the anomaly survey by Chandola, Banerjee, and Kumar distinguishes between point, contextual and collective anomalies [28]. Feature selection studies for IoT intrusion detection involve population-based search and ensemble evaluation. Abu Alghanam et al. use a pigeon-inspired optimizer with ensemble learning for IoT network intrusion detection [29]. Prazeres et al. evaluate design choices for an ML-based flow IDS and show that operational data preparation influences the final system [30]. In this paper, we use a deterministic filter score before classifier fitting. This design avoids the overhead of repeatedly training a wrapper on every candidate subset and explicitly decomposes class separation, overlap, rarity, and redundancy.

C. Outlier Detection and Data Quality

Outlier detection is concerned with finding samples that are significantly different from an assumed source distribution. Isolation Forest uses short random partition paths to define anomaly scores [27]. The anomaly survey by Chandola, Banerjee and Kumar differentiates point, contextual and collective anomalies and describes how definition of an anomaly is dependent on application context [28]. In cyber attacks, an attack can be an anomaly by design. Blind subtraction produces a circular detector which destroys the rarest attacks, and then the trained classifier accurately reports the residual traffic.

MFRR-CEF decouples structural invalidity from behavioral rarity. A structural rule can correct a negative duration, an impossible number of packets, a malformed categorical value, or an exact duplicate generated by processing. A strong statistical rule may clip a corrupted tail or tag a flow for inspection. A class-conditional defense rule preserves extreme minority flows unless they break a structural rule or they are unsupported within the class neighborhood. This separation is central to the first algorithm.

Scaling may also reveal information. Median, interquartile range, clipping limits, mappings for categorical types and rules for missing-values should be computed on a training fold and then applied as is to its validation or test fold. When you run preprocessing on a full dataset, you are looking at distributional information that includes out-of-sample traffic. Duplicates or flows from a single session may also cross a random row split. Group identifier: when metadata allow, a group identifier should keep connected flows, capture windows, source hosts or scenario runs together in one partition.

D. IoT Intrusion-Detection Models

Deep anomaly systems learn nonlinear flow representations. Saba et al. developed an anomaly-based deep intrusion detector for IoT networks [31], and Sharma et al. tested deep learning for IoT attack detection [32]. Deep models can represent complex interactions but may need more training data, hyperparameter tuning, and computational resources for inference than a gateway can provide. They have reported accuracy related to a specific dataset, split, feature set and label taxonomy.

Stacking is a method that combines multiple learners' predictions. Lazzarini, Tianfield and Charissis apply a deep stacking ensemble for IoT intrusion detection [33]. Cao et al. they also propose a stacked ensemble for iot network traffic [34]. Stacking can learn that one model is more trustworthy, but the meta-learner must be trained on out-of-fold predictions. Training it on the same fitted predictions as the base learners, however, produces an over-optimistic estimate. MFRR-CEFlinearly restricts fusion fitting to the inner validation outputs.

More attention has been paid to explainability and feature analysis. Muna et al. investigate explanation techniques for IoT attack detection in smart city environments [35]. Syed, Ge, and Baig integrated feature selection with recurrent learning for a fog-cloud IDS [36]. Kornyo et al. used recursive feature elimination for botnet classification [37]. BotStop is a packet-level ML approach with an explainability objective [38]. Sarhan et al. investigated feature selection for ML-based IoT NIDS and found that representation affects both detection and generalization [39]. MFRR+ is explainable at the score level, where each chosen feature has a pairwise separation, overlap, rarity, and redundancy component. A local decision explanation also requires a classifier-specific approach.

Generative balancing has been used with IoT traffic. Habibi, Chemmakha and Lazaar employ CTGAN along with machine learning for imbalanced IoT botnet data [40]. A generative model can also represent mixed tabular distributions, but its validation requires semantic constraints and privacy checks. Guarded interpolation in MFRR-CEF is more conservative: it generates candidates within minority neighborhoods and reject candidates that cross a local decision margin.

Contemporary hybrid models integrate representation learning, synthesis and ensembles. HDA-IDS employs semi-supervised c-learning to detect DoS [41]. Honey-block integrates edge support, ensemble learning and preventive control [42]. Adaptive model selection for early IoT bot detection has been considered [43]. The IoTBoT-IDS approach leverages statistical learning in smart- city botnet detection [44], and Distributed Anomaly Detection (DAD) employs distributed one-class statistical ensembles on the edge [45]. These are the good designs from the exhibits. They also stress the need to declare whether a model is binary anomaly detection, multiclass attack labelling, distributed adaptation, or active response. MFRR-CEF is designed to perform offline multiclass flow labelling with an abstention output, and the authors do not advocate for it to block traffic autonomously.

E. Calibration, Selective Output, and Explanation

Modern classifiers can be accurate and underconfident (poorly calibrated). Temperature scaling learns a scalar towards the validation logits and commonly decreases the expected calibration error without influencing the top class [46]. Feature attribution can support error examination. SHAP represents a prediction in terms of additive

feature contributions with the background chosen distribution [47]. Its values depend on the explainer, dependence assumption and reference sample. A large attribution does not imply a causal mode of attack. The reporting scheme combines global MFRR+ scores, permutation tests on held-out folds, with local SHAP explanations of confirmed and unconfirmed flows. Address-like identifiers are removed so their attributions don't dominate a report.

SVM pairwise probability estimation The pairwise probability estimation maps margins into pairwise probabilities and couples them into a multiclass vector [48], [49]. Tree ensembles estimate class frequencies at leaves or over trees, however those frequencies may also require calibration. MFRR-CEF calibrates each branch based on out-of-fold predictions and runs a second calibration check after fusion.

Selective outputs yields better auto-error, but at the cost of coverage. The final decision rule accepts a class only if the calibrated confidence is greater than a validation threshold and the difference between the top-two probabilities is large enough. Low-confidence flows are bottled into a review queue. This criterion is tested on risk-coverage curves in an offline benchmark. In a deployment, it must hook into rate limits, analyst capacity, and fallback rules.

F. Research Gap

Literature provides strong pieces but rarely brings them together under a single evidence boundary. Feature-selection papers may or may not rebalance before splitting. Ensemble papers may or may not consider calibration and diversity. Dataset papers show shared NetFlow attributes aid comparison but also show that multiclass performance is driven by taxonomy and traffic source. The open problem is an interpretable fold-local framework that integrates class support, pairwise overlap, redundant attributes, complementary classifiers, and calibrated output simultaneously in a unified framework.

MFRR-CEF is a score and processing chain for the multiclass problem. MFRR+ pairwise scoring explicitly assigns a role to each class pair. Guarded synthesis uses local support rather than a global duplication target. NSVC and ERF are two different boundaries. Fusion uses validation quality and error diversity. The paper provides the metrics that are available in the literature in the experimental section and defines the missing paired analysis without creating any artificial data.

III. Problem Formulation and Research Design

A. Data Model and Labels

Let a NetFlow dataset be

$$\mathcal{D} = \{(\boldsymbol{x}_i, y_i, g_i, t_i)\}_{i=1}^{N}, \qquad (1)$$

where $\boldsymbol{x}_i \in \mathbb{R}^{d_n} \times \mathcal{A}^{d_c}$ contains numeric and categorical flow attributes, $y_i \in \{1, \dots, C\}$ is a benign or attack label, g_i is a group identifier for a capture, session, host, or scenario, and t_i is a timestamp when available. The group variable is used for splitting but is not a predictor. A preprocessing map T_k is fitted only from the training portion of outer fold k,

$$\tilde{\boldsymbol{x}}_i = T_k(\boldsymbol{x}_i), \qquad i \in \mathcal{I}_{k,tr} \cup \mathcal{I}_{k,te}. \qquad (2)$$

The fitted parameters of T_k include categorical maps, imputation values, clipping limits, medians, scales, selected features, and any sampling state. Test labels do not enter T_k.

The class prior in a training fold is

$$\pi_c = \frac{n_c}{n}, \qquad n_c = |\{i \in \mathcal{I}_{tr} : y_i = c\}|. \qquad (3)$$

The imbalance ratio relative to class c is $\rho_c = max_r n_r / n_c$. The vector $\boldsymbol{\rho}$ is more informative than one majority-to-minority scalar in a multiclass corpus.

B. Leakage Boundary

For each outer fold, the training and test group sets satisfy

$$\{g_i : i \in \mathcal{I}_{k,tr}\} \cap \{g_i : i \in \mathcal{I}_{k,te}\} = \varnothing. \qquad (4)$$

Hyperparameters are selected by grouped inner folds contained in $\mathcal{I}_{k,tr}$. Let $\mathcal{V}_{k,h}$ be inner validation fold h and $\mathcal{R}_{k,h}$ its inner training complement. The selected configuration is

$$\widehat{\boldsymbol{\theta}}_k = \underset{\boldsymbol{\theta} \in \Theta}{argmax} \frac{1}{H} \sum_{h=1}^{H} MacroF1\left(\mathcal{V}_{k,h}; Fit\left(\mathcal{R}_{k,h}, \boldsymbol{\theta}\right)\right). \quad (5)$$

The outer test fold is evaluated once using $\widehat{\boldsymbol{\theta}}_k$. This nested design isolates model selection from performance estimation.

C. Outputs and Metrics

For a flow $\boldsymbol{x}$, the system returns

$$(\hat{y}, \hat{\boldsymbol{p}}, q) = F(\boldsymbol{x}), \quad (6)$$

where $\hat{\boldsymbol{p}} \in [0,1]^C$, $\sum_c \hat{p}_c = 1$, and $q \in \{\text{accept,review}\}$. Class c has true-positive, false-positive, and false-negative counts TP_c, FP_c, and FN_c. Its precision, recall, and F1-score are

$$P_c = \frac{TP_c}{TP_c+FP_c}, \quad R_c = \frac{TP_c}{TP_c+FN_c}, \quad F_c = \frac{2P_cR_c}{P_c+R_c}. \quad (7)$$

Macro F1 is

$$F_{macro} = \frac{1}{C}\sum_{c=1}^{C} F_c. \quad (8)$$

This value assigns equal weight to every label. Accuracy is retained for continuity with the comparative experiment,

$$A = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}\,(\hat{y}_i = y_i), \quad (9)$$

but it is not the only criterion for model selection. Per-class precision-recall area is reported for rare labels, since precision-recall curves are more informative than receiver-operating-characteristic curves when the negative class is significantly larger [50]. The multiclass Matthews correlation coefficient is reported as a confusion-matrix summary with sensitivity to all cells [51].

Calibration is measured by log loss and expected calibration error. With bins B_m,

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{N} |acc(B_m) - conf(B_m)|. \quad (10)$$

For a confidence threshold τ, coverage and selective risk are

$$Cov(\tau) = \frac{1}{N}\sum_i \mathbb{I}\left(\max_c \hat{p}_{ic} \geq \tau\right), \quad (11)$$

$$Risk(\tau) = \frac{\sum_i \mathbb{I}(\hat{y}_i \neq y_i)\mathbb{I}(q_i = accept)}{\sum_i \mathbb{I}(q_i = accept)}. \quad (12)$$

D. Hypotheses

The complete benchmark evaluates five directional hypotheses. H_1 states that MFRR+ improves outer-fold macro F1 over an unfiltered feature set. H_2 states that guarded balancing improves the mean recall of the three least frequent classes over class weighting alone. H_3 states that entropy-weighted NSVC improves minority-class recall over an unweighted SVC. H_4 states that supervised ERF improves macro F1 over ordinary Random Forest and ordinary Rotation Forest. H_5 states that calibrated diversity fusion improves macro F1 or log loss over its strongest component without increasing false positives beyond a prespecified tolerance.

Such hypotheses need paired outer-fold or row-level predictions. The aggregate plots facilitate descriptive comparisons. The correct counts of the pilots support one-sample binomial intervals. They do not specify which Pilot A is paired with which Pilot B, hence a paired McNemar test cannot be reconstructed.

E. Design Constraints

The model is subject to eight constraints. Raw source and destination addresses are not predictors. Split groups are determined prior to preprocessing. Outlier treatment cannot remove a true minority flow if it is only rare. Each chosen feature is assigned a decomposable score. Synthetic samples must satisfy a competing-class margin test. Class costs are constrained. Fusion weights are non-negative and sum to one. Calibration and review thresholds are estimated using inner-test labels only.

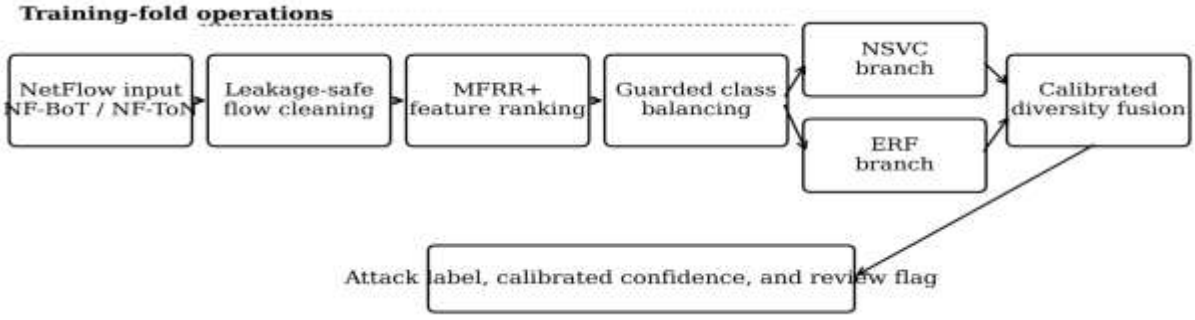


Fig. 1. Architecture of MFRR-CEF. Flow preparation, MFRR+ ranking and guarded balancing are all performed in the training fold. Entropy-weighted NSVC and supervised ERF are two complementary branches. Calibrated fusion outputs an attack label, confidence vector, and a review flag.

IV. Proposed MFRR-CEF Framework

A. Leakage-Safe Flow Preparation

The preparation phase checks schema, labels, ranges, and group ids. Direct identifiers, raw IP addresses, record indices and fields derived from target label are deleted. Ports and protocol designators may be left in when they are known at operational time but a dataset-specific port must be tested in temporal and cross-dataset experiments. Duplicate flows are clustered prior to partitioning. When scenario or host identifiers are available, all flows belonging to a group remain in a single outer fold of cross-validation.

For numeric feature j, the training median and median absolute deviation are

$$m_j = median_{i\in\mathcal{I}_{tr}} x_{ij}, \qquad a_j = median_{i\in\mathcal{I}_{tr}} \left|x_{ij} - m_j\right|. \quad (13)$$

The robust scale is $s_j = 1.4826a_j + \epsilon$. A standardized value is

$$z_{ij} = clip\left(\frac{x_{ij}-m_j}{s_j}, -b_j, b_j\right), \quad (14)$$

where b_j is selected from training quantiles or a fixed operational limit. Missing numeric values are assigned m_j and accompanied by a missingness indicator when missingness can carry process information. Categorical values are mapped to an explicit vocabulary plus an unknown token.

A class-conditional robust distance is

$$d_{ic}^2 = \sum_{j\in\mathcal{J}} \left(\frac{x_{ij}-m_{jc}}{s_{jc}}\right)^2, \quad (15)$$

where m_{jc} and s_{jc} are computed from class c inside the training fold. A sample can be flagged when d_{i,y_i} exceeds a class quantile. The protection condition retains the sample if it has at least k_p same-class neighbors inside radius r_p and violates no structural rule. This rule treats rarity as possible attack evidence.

B. MFRR+ Pairwise Feature Scoring

For feature j and class pair (a, b), standardized location separation is

$$\Delta_{j,ab} = \frac{|\mu_{j,a}-\mu_{j,b}|}{\sqrt{\sigma_{j,a}^2+\sigma_{j,b}^2+\epsilon}}. \quad (16)$$

Let $\hat{f}_{j,a}$ and $\hat{f}_{j,b}$ be training-fold marginal density estimates on a common grid. Their overlap coefficient is

$$O_{j,ab} = \int min\{\hat{f}_{j,a}(u), \hat{f}_{j,b}(u)\}\, du, \quad (17)$$

with $0 \le O_{j,ab} \le 1$. A quantile-overlap estimator replaces the density integral when either class has fewer than the minimum density-support count.

Class entropy is

$$H(\boldsymbol{\pi}) = -\sum_{c=1}^{C} \pi_c\, log(\pi_c + \epsilon). \quad (18)$$

The normalized rarity weight for pair (a, b) is

$$\omega_{ab} = \frac{min\{\omega_{max}, (n_a+\epsilon)^{-1/2}+(n_b+\epsilon)^{-1/2}\}}{\sum_{r<s} min\{\omega_{max}, (n_r+\epsilon)^{-1/2}+(n_s+\epsilon)^{-1/2}\}}. \quad (19)$$

The raw pairwise score is

$$U_j = \frac{H(\boldsymbol{\pi})}{logC}\sum_{a<b} \omega_{ab}\ \psi\left(\Delta_{j,ab}\right)\left(1 - O_{j,ab}\right), \quad (20)$$

where $\psi(u) = u/(1+u)$ bounds univariate separation in $[0,1)$. Let ρ_{jk} be a Spearman correlation on training data. The redundancy burden relative to already retained set $\mathcal{S}$ is

$$R_j = \begin{cases} 0, & \mathcal{S} = \varnothing, \\ |\mathcal{S}|^{-1}\sum_{k\in\mathcal{S}} max\left(0, \left|\rho_{jk}\right| - \tau_\rho\right), & \text{otherwise.} \end{cases} \quad (21)$$

MFRR+ greedily selects the feature with maximum

$$S_j = \frac{U_j}{1+\lambda_R R_j} \quad (22)$$

Query until the desired count, validation plateau, or cost budget is reached. The score is still decomposable by class pairs and tells us why a feature is kept.

C. Boundary-Guarded Class Balancing

The balancing stage combines bounded class weights with optional synthetic support. Its target for class c is

$$n_c^\star = min\{n_{max}, \lceil \alpha n_{med} + (1-\alpha)n_c \rceil\}, \quad (23)$$

where n_{med} is the median class count and $0 \le \alpha \le 1$. The target does not force every class to the majority count. A candidate between minority samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_r$ is

$$\boldsymbol{x}' = \boldsymbol{x}_i + \lambda(\boldsymbol{x}_r - \boldsymbol{x}_i), \qquad \lambda \sim \mathcal{U}(0,1). \tag{24}$$

Let $d_c(\boldsymbol{x}')$ be the average distance to the k nearest class-c neighbors and $d_{\neg c}(\boldsymbol{x}')$ the average distance to the nearest competing-class neighbors. The local safety margin is

$$M_c(\boldsymbol{x}') = d_{\neg c}(\boldsymbol{x}') - d_c(\boldsymbol{x}'). \tag{25}$$

The candidate is accepted only if $M_c(\boldsymbol{x}') \ge \delta_c$, all structural constraints are satisfied, and each categorical field takes a permitted training value. Rejected candidates are not replaced by arbitrary duplicates after a retry limit. The classifier also receives a bounded class cost,

$$w_c = clip\left[\left(\frac{n}{Cn_c}\right)^{\eta}, w_{min}, w_{max}\right]. \tag{26}$$

This combination adds local support without allowing a tiny class to produce an unbounded loss coefficient.

D. Entropy-Weighted NSVC

For class pair (a, b), NSVC solves

$$\min_{\boldsymbol{w}_{ab},\beta_{ab},\xi} \frac{1}{2} \| \boldsymbol{w}_{ab} \|_2^2 + C \sum_{i:y_i \in \{a,b\}} w_{y_i} \xi_i, \tag{27}$$

subject to

$$\tilde{y}_{i,ab}[\boldsymbol{w}_{ab}^{\top}\phi(\boldsymbol{x}_i) + \beta_{ab}] \ge 1 - \xi_i, \quad \xi_i \ge 0. \tag{28}$$

Here $\tilde{y}_{i,ab} \in \{-1, +1\}$. Pair importance is

$$e_{ab} = \frac{-p_{a|ab}logp_{a|ab} - p_{b|ab}logp_{b|ab}}{log2}, \tag{29}$$

where $p_{a|ab} = n_a/(n_a + n_b)$. A pair with balanced support has high entropy; an extreme pair receives class costs from (26). Validation calibration maps margin $f_{ab}(\boldsymbol{x})$ to pairwise probability

$$r_{ab}(\boldsymbol{x}) = [1 + exp(A_{ab}f_{ab}(\boldsymbol{x}) + B_{ab})]^{-1}. \tag{30}$$

Pairwise coupling obtains a probability vector $\boldsymbol{p}^{(S)}$ that is consistent with the set of r_{ab} values [48], [49].

E. Supervised Enhanced Rotation Forest

The selected feature set is divided into G groups. Ordinary random grouping is replaced by a constrained allocation that limits within-group correlation and preserves high MFRR+ features across groups. For group g, a class-stratified bootstrap matrix $\boldsymbol{X}_g$ is centered and decomposed,

$$\boldsymbol{X}_g^{\top}\boldsymbol{X}_g = \boldsymbol{V}_g\boldsymbol{\Lambda}_g\boldsymbol{V}_g^{\top}. \tag{31}$$

For component q, the discriminative ratio is

$$J_{gq} = \frac{\boldsymbol{v}_{gq}^{\top}\boldsymbol{S}_{B,g}\boldsymbol{v}_{gq}}{\boldsymbol{v}_{gq}^{\top}(\boldsymbol{S}_{W,g} + \epsilon\boldsymbol{I})\boldsymbol{v}_{gq}}, \tag{32}$$

where $\boldsymbol{S}_{B,g}$ and $\boldsymbol{S}_{W,g}$ are between- and within-class scatter matrices. Components are retained by a combined variance and J_{gq} rule. Their orthonormal columns form group rotation $\boldsymbol{R}_g$, and a block-diagonal matrix forms the complete rotation,

$$\boldsymbol{R} = blockdiag(\boldsymbol{R}_1, \dots, \boldsymbol{R}_G). \tag{33}$$

Each balanced tree is trained on $\boldsymbol{z}_i = \boldsymbol{R}^{\top}\boldsymbol{x}_i$. Repetition with different bootstraps and groupings produces ERF posterior $\boldsymbol{p}^{(R)}$.

F. Calibrated Diversity Fusion

NSVC and ERF are fitted on inner training data, and their predictions on inner validation folds form out-of-fold probability matrices. Let L_m be multiclass log loss for branch m, F_m its macro F1, and D_m its average disagreement with the other branch on cases where at least one branch is correct. A branch utility is

$$u_m = \gamma_F F_m - \gamma_L L_m + \gamma_D D_m. \tag{34}$$

Nonnegative fusion weights are

$$\alpha_m = \frac{exp(u_m/T_w)}{\sum_r exp(u_r/T_w)}, \qquad \sum_m \alpha_m = 1. \tag{35}$$

The fused probability is

$$\boldsymbol{p} = \alpha_S\boldsymbol{p}^{(S)} + \alpha_R\boldsymbol{p}^{(R)}. \tag{36}$$

A temperature or vector-scaling map is fitted to the fused out-of-fold predictions. The decision rule is

$$q(\boldsymbol{x}) = \begin{cases} accept, & p_{(1)} \ge \tau_p \ \wedge \ p_{(1)} - p_{(2)} \ge \tau_g, \\ review, & \text{otherwise}, \end{cases} \tag{37}$$

where $p_{(1)}$ and $p_{(2)}$ are the largest two calibrated probabilities. Thresholds τ_p and τ_g are selected from inner data according to a declared coverage or review-capacity target.

G. End-to-End Training and Inference

The entire model is fitted separately in each outer training fold. Flow-cleaning parameters are estimated first, then MFRR+ scores. The inner loop chooses the number of features, density bandwidth, redundancy penalty, balancing target, NSVC kernel and costs, ERF tree number and grouping, fusion temperature, and review thresholds. Once selected, the entire outer training fold is transformed, balanced, and fitted a single time. Outer test flows are given the same transform and have no synthetic samples. This way we keep the test distribution outside every learned design choice.

V. ALGORITHMS AND MATHEMATICAL ANALYSIS

A. Algorithm 1: Leakage-Safe Robust Flow Preparation

Algorithm 1 Leakage-Safe Robust Flow Preparation (LSRFP)

Input: labeled training flows $\mathcal{D}_{tr}$; unlabeled evaluation flows $\mathcal{D}_{ev}$; group field g; protected class support k_p; clipping level b.

Output: fitted transformation T; cleaned training matrix $\boldsymbol{Z}_{tr}$; transformed evaluation matrix $\boldsymbol{Z}_{ev}$; processing log $\mathcal{L}$.

Line	Operation
1	Validate the schema, label vocabulary, numeric domains, categorical domains, and group identifier.
2	Remove record IDs, raw addresses, direct label proxies, post-event fields, and columns unavailable at inference.
3	Form duplicate and session groups before any split; reject a partition that divides one group.
4	Fit numeric medians m_j, robust scales s_j, and clipping limits on $\mathcal{D}_{tr}$ only.
5	Fit categorical vocabularies with one unknown token and fit missing-value indicators on training data.
6	For each training sample i, test structural constraints; remove the row if any hard constraint fails.
7	Compute class-conditional distance d_{i,y_i}; flag a statistical outlier above its training quantile.
8	If a flagged row has k_p same-class neighbors and no hard failure, retain it as a protected rare pattern.
9	Impute, clip, and scale retained training rows; apply the stored map unchanged to evaluation rows.
10	Record every excluded column, removed row, clipped value, unknown category, and fitted parameter in $\mathcal{L}$.
11	Return T, $\boldsymbol{Z}_{tr}$, $\boldsymbol{Z}_{ev}$, and $\mathcal{L}$.

Complexity: $O(nd + nk_p d)$ with an indexed neighbor search; $O(n^2 d)$ for an exhaustive search. Storage is $O(nd)$.

Safeguard: no statistic is refitted on evaluation data, and statistical rarity alone cannot delete a supported minority sample.

Link to ablation: substitute LSRFP with global mean imputation and z-score scaling; evaluate leakage indicators, macro F1, recall for the minority class, and performance across datasets.

Proposition 1 (fold isolation and bounded transform). If all parameters of T_k are measurable functions of the outer training fold and every numeric value is clipped to $[-b_j, b_j]$, the transformed test feature is bounded and independent of outer-test labels conditional on the fitted training map.

Proof: Let $\mathcal{F}_{k,tr}$ be the information generated by the training fold. The fitted median, scale, category map, outlier limits, and selected column set are $\mathcal{F}_{k,tr}$-measurable. For any test row and feature j,

$$\begin{aligned} m_{jk} &= m_j(\mathcal{D}_{k,tr}), \\ s_{jk} &= s_j(\mathcal{D}_{k,tr}) + \epsilon > 0, \\ u_{ij} &= (x_{ij} - m_{jk})/s_{jk}, \\ z_{ij} &= max\{-b_j, min(b_j, u_{ij})\}, \\ -b_j &\le z_{ij} \le b_j, \\ |z_{ij}| &\le b_j, \\ \| \boldsymbol{z}_i \|_2^2 &= \sum_{j=1}^{d} z_{ij}^2 \le \sum_{j=1}^{d} b_j^2 , \\ \| \boldsymbol{z}_i \|_2 &\le \| \boldsymbol{b} \|_2, \\ T_k &\perp\!\!\!\perp \{y_i : i \in \mathcal{I}_{k,te}\} \mid \mathcal{F}_{k,tr}. \end{aligned} \tag{38}$$

The norm bound limits the influence of an extreme numeric value on distance and kernel calculations. The conditional independence follows from the algorithmic restriction that outer-test labels never enter the fitted map. Group-disjoint splitting also prevents an identical session from defining both the fitted map and its evaluation. The proposition does not claim independence between real traffic distributions; it establishes procedural isolation. ▫

B. Algorithm 2: MFRR+ Feature Ranking

Algorithm 2 Multi-Class Feature Rank Ratio Plus (MFRR+)

Input: cleaned training matrix $\boldsymbol{Z} \in \mathbb{R}^{n\times d}$; labels $\boldsymbol{y}$; desired maximum K; overlap support n_0; correlation limit τ_ρ; redundancy coefficient λ_R.

Output: ordered feature list $\mathcal{S}$; total scores S_j; pairwise score ledger Q.

Line	Operation
1	Compute training class counts, priors π_c, normalized entropy, and capped pair weights ω_{ab}.
2	For each feature j and each class pair (a,b), compute $\mu_{j,c}$, $\sigma_{j,c}^2$, and $\Delta_{j,ab}$.
3	Estimate density overlap $O_{j,ab}$ on a common training grid; use quantile overlap when support is below n_0.
4	Store $q_{j,ab} = \omega_{ab}\psi(\Delta_{j,ab})(1 - O_{j,ab})$ in ledger Q.
5	Sum pair contributions and multiply by normalized class entropy to obtain raw score U_j.
6	Initialize $\mathcal{S} \leftarrow \varnothing$.
7	For selection step $r = 1, \dots, K$, compute redundancy R_j against features already in $\mathcal{S}$.
8	Select $j^\star = argmax_{j\notin\mathcal{S}} U_j/(1 + \lambda_R R_j)$ and append it to $\mathcal{S}$.
9	Stop when inner-fold macro F1 fails to improve by δ_F for p consecutive additions or when cost exceeds budget.
10	Normalize final S_j by the maximum selected score and return $\mathcal{S}$, $\boldsymbol{S}$, and Q.

Complexity: density estimation and pair scoring require $O(C^2 d n_g + nd)$ for a grid of n_g points; correlation calculation is $O(nd^2)$ before optional screening. Greedy selection is $O(KdK)$.

Safeguard: every density, correlation, stop rule, and feature count is learned inside the inner training process.

Ablation link: remove overlap, entropy, rarity, or redundancy one term at a time; report rank stability, selected-feature count, macro F1, and the recall of each rare class.

Proposition 2 (boundedness and monotonic pair contribution). Under normalized densities, positive finite class counts, and $\lambda_R \geq 0$, every MFRR+ score lies in $[0,1]$. For fixed overlap and weights, a pair contribution is nondecreasing in standardized separation; for fixed separation and weights, it is nonincreasing in overlap.

Proof: From normalized densities and the definition of overlap,

$$\begin{aligned}
0 &\leq min\{\hat{f}_{j,a}(u), \hat{f}_{j,b}(u)\}, \\
0 &\leq O_{j,ab} = \int min\{\hat{f}_{j,a}, \hat{f}_{j,b}\}\, du \leq 1, \\
0 &\leq 1 - O_{j,ab} \leq 1, \\
\Delta_{j,ab} &\geq 0, \\
\psi(\Delta) &= \Delta/(1+\Delta) \in [0,1), \\
\psi'(\Delta) &= 1/(1+\Delta)^2 > 0, \\
\textstyle\sum_{a<b} \omega_{ab} &= 1, \quad \omega_{ab} \geq 0, \\
0 &\leq H(\boldsymbol{\pi})/logC \leq 1, \\
0 &\leq U_j \leq 1, \\
R_j &\geq 0 \Rightarrow 1 + \lambda_R R_j \geq 1, \\
0 &\leq S_j = U_j/(1 + \lambda_R R_j) \leq 1, \\
\partial q_{j,ab}/\partial \Delta &= \omega_{ab}(1 - O_{j,ab})/(1+\Delta)^2 \geq 0, \\
\partial q_{j,ab}/\partial O &= -\omega_{ab}\psi(\Delta) \leq 0.
\end{aligned} \tag{39}$$

The bounded map prevents a feature with an extreme variance ratio from overwhelming the complete ranking. The derivatives establish the intended direction of both pairwise terms. A redundancy burden can only decrease the score, so the same upper bound remains valid after greedy selection. ▫

C. Algorithm 3: Boundary-Guarded Multiclass Balancing

Algorithm 3 Boundary-Guarded Multiclass Balancing (BGMB)

Input: selected training data $(\boldsymbol{Z}, \boldsymbol{y})$; target parameter α; neighbor count k; class margin δ_c; retry limit r_{max}; class-cost exponent η.

Output: balanced training set $(\boldsymbol{Z}^{\star}, \boldsymbol{y}^{\star})$; bounded class costs w_c; acceptance log $\mathcal{B}$.

Line	Operation
1	Compute class counts, the median count, bounded targets $n_c^{\star}$, and costs w_c.
2	Build a same-class neighbor index and a competing-class neighbor index in the selected scaled space.
3	For each class c with $n_c < n_c^{\star}$, select an anchor from low-support regions with capped probability.
4	Select one of the anchor's k nearest same-class neighbors and interpolate numeric attributes.
5	Draw categorical fields from the anchor-neighbor pair or an observed local conditional distribution.
6	Reject a candidate that violates a protocol, range, count, flag, or feature-consistency constraint.
7	Compute $M_c(\boldsymbol{x}') = d_{\neg c}(\boldsymbol{x}') - d_c(\boldsymbol{x}')$.
8	Accept the candidate only when $M_c(\boldsymbol{x}') \geq \delta_c$ and its nearest label is c.
9	Stop a class at its target or retry limit; never fill a failed target with unconstrained duplication.
10	Return the augmented set, bounded costs, acceptance rate, rejection causes, and local-margin distribution.

Complexity: neighbor indexing is $O(ndlogn)$ for a tree structure; generation is $O(n_s kd)$ for n_s accepted and attempted samples. High-dimensional exact search can approach $O(n_s nd)$.

Safeguard: synthesis is restricted to an inner training fold, categorical combinations must be either observed or allowed, and each accepted candidate has a positive competing-class margin.

Ablation link: no sampling and future prediction with ordinary SMOTE, class cost, BGMB only and BGMB plus class cost under the same outer folds.

Proposition 3 (local label preservation). If a candidate $\boldsymbol{x}'$ is accepted with $M_c(\boldsymbol{x}') \geq \delta_c > 0$, then its mean same-class neighbor distance is at least δ_c smaller than its mean competing-class distance. For a nearest-centroid or locally Lipschitz score with bounded perturbation, the accepted candidate remains in the class-c local region.

Proof: By the acceptance rule,

$$\begin{aligned}
M_c(\boldsymbol{x}') &= d_{\neg c}(\boldsymbol{x}') - d_c(\boldsymbol{x}') \geq \delta_c, \\
d_{\neg c}(\boldsymbol{x}') &\geq d_c(\boldsymbol{x}') + \delta_c, \\
d_c(\boldsymbol{x}') &< d_{\neg c}(\boldsymbol{x}'), \\
\hat{y}_{local}(\boldsymbol{x}') &= c, \\
|d_c(\boldsymbol{x}' + \boldsymbol{e}) - d_c(\boldsymbol{x}')| &\leq \| \boldsymbol{e} \|_2, \\
|d_{\neg c}(\boldsymbol{x}' + \boldsymbol{e}) - d_{\neg c}(\boldsymbol{x}')| &\leq \| \boldsymbol{e} \|_2, \\
M_c(\boldsymbol{x}' + \boldsymbol{e}) &\geq M_c(\boldsymbol{x}') - 2 \| \boldsymbol{e} \|_2, \\
\| \boldsymbol{e} \|_2 &< \delta_c/2, \\
M_c(\boldsymbol{x}' + \boldsymbol{e}) &> 0, \\
\hat{y}_{local}(\boldsymbol{x}' + \boldsymbol{e}) &= c.
\end{aligned} \tag{40}$$

The distance functions are 1-Lipschitz with respect to a common metric when defined as means of distances to fixed neighbor sets. Any perturbation smaller than half the accepted margin retains positive local separation. The result does not prove that a synthetic flow is physically generated by a real device; the structural constraints and held-out tests remain necessary. ▫

D. Algorithm 4: Entropy-Weighted NSVC

Algorithm 4 Entropy-Weighted New Support Vector Classifier (E-NSVC)

Input: balanced selected data; bounded class costs w_c; kernel set $\mathcal{K}$; penalty grid $\mathcal{C}$; grouped inner folds.

Output: pairwise margin models; calibrated pair probabilities; coupled posterior $\boldsymbol{p}^{(S)}$.

Line	Operation
1	For each class pair (a, b), extract only the two classes and compute pair entropy e_{ab}.
2	Select kernel and penalty using inner-fold macro F1 and log loss, subject to bounded class costs.

Line	Operation
3	Fit the weighted soft-margin problem for pair (a, b).
4	Generate inner out-of-fold margins for every sample in the pair.
5	Fit monotone sigmoid parameters (A_{ab}, B_{ab}) on the out-of-fold margins.
6	At inference, compute every pair margin and calibrated probability r_{ab}.
7	Couple pair probabilities into $\boldsymbol{p}^{(S)}$ and normalize numerical residuals.
8	Return the maximum-posterior class, full posterior, margin gap, and support-vector count.

Complexity: kernel SVC training ranges from approximately $O(n_{ab}^2 d)$ to $O(n_{ab}^3)$ per pair; inference is $O(C^2 n_{sv} d)$. Linear solvers reduce cost for large flow sets.
Safeguard: calibration uses out-of-fold margins, class costs are clipped, and kernel selection never sees the outer test fold.
Ablation link: compare ordinary SVC, inverse-frequency SVC, entropy-weighted NSVC without calibration, and complete E-NSVC.
Proposition 4 (minority violation influence under bounded costs). For a pair (a, b), if $n_a < n_b$, $\eta > 0$, and neither class weight reaches a clipping boundary, each class-a margin violation receives a larger coefficient than each class-b violation. Clipping gives a finite upper bound on the complete empirical hinge term.
Proof: From (26), before clipping,

$$\begin{aligned} \widetilde{w}_a &= \left(n/(Cn_a)\right)^{\eta}, \\ \widetilde{w}_b &= \left(n/(Cn_b)\right)^{\eta}, \\ n_a < n_b &\Rightarrow n/n_a > n/n_b, \\ \eta > 0 &\Rightarrow \widetilde{w}_a > \widetilde{w}_b, \\ w_a &= \widetilde{w}_a > w_b = \widetilde{w}_b, \\ \ell_i &= C w_{y_i} max(0, 1 - \tilde{y}_i f_i), \\ \partial \ell_i / \partial \xi_i &= C w_{y_i}, \\ C w_a &> C w_b, \\ w_{min} &\le w_c \le w_{max}, \\ 0 \le \textstyle\sum_i C\, w_{y_i} \xi_i &\le C w_{max} \textstyle\sum_i \xi_i . \end{aligned} \tag{41}$$

The loss assigns greater marginal cost to an uncorrected rare-class violation whenever the weights are internal to the clipping range. The finite bound avoids the unbounded coefficient produced by raw inverse frequency at extremely small n_c. At a clipping boundary the ordering can become equal, but it cannot reverse under a monotone clip. ▫

E. Algorithm 5: Supervised Enhanced Rotation Forest

Algorithm 5 Supervised Enhanced Rotation Forest (S-ERF)
Input: selected and balanced training data; number of trees B; feature groups G; retained-component rule κ; tree controls.
Output: tree ensemble; class posterior $\boldsymbol{p}^{(R)}$; rotation ledger.

Line	Operation
1	Distribute high MFRR+ features across G groups and minimize within-group absolute correlation.
2	For tree $b = 1, \ldots, B$, draw a class-stratified bootstrap with capped minority duplication.
3	For each group g, center its data and compute covariance eigenvectors.
4	Compute Fisher ratio J_{gq} for each eigenvector using class scatter matrices.
5	Retain components meeting the cumulative-variance rule or the discriminative-ratio rule.
6	Orthonormalize retained vectors and assemble block-diagonal rotation $\boldsymbol{R}^{(b)}$.
7	Transform training data and fit a class-balanced decision tree with prespecified depth and leaf limits.
8	Store the bootstrap seed, feature grouping, retained components, rotation, and tree parameters.
9	At inference, rotate the flow for each tree and average leaf probability vectors.
10	Return $\boldsymbol{p}^{(R)}$ and per-tree votes.

Complexity: group eigendecomposition costs $O\left(B\sum_g d_g^3\right)$ plus $O(Bndlogn)$ for tree induction; inference is $O(Bd + B\ell)$ for tree path length ℓ.

Safeguard: every rotation is fitted on one training bootstrap; orthonormalization is checked numerically; depth and leaf size limit memorization.

Ablation link: compare Random Forest, ordinary Rotation Forest, supervised grouping alone, Fisher-guided retention alone, and complete S-ERF.

Proposition 5 (distance preservation by full orthonormal rotation). Suppose now that each group rotation is an orthonormal matrix, and that all group dimensions are preserved. Then, the block-diagonal rotation preserves Euclidean distances and total variance. Truncation of components is nonexpansive.

Proof: Let $\boldsymbol{R} = blockdiag(\boldsymbol{R}_1, \dots, \boldsymbol{R}_G)$. Then

$$\begin{aligned}
\boldsymbol{R}_g^\top \boldsymbol{R}_g &= \boldsymbol{I}_{d_g}, \\
\boldsymbol{R}^\top \boldsymbol{R} &= blockdiag\left(\boldsymbol{I}_{d_1}, \dots, \boldsymbol{I}_{d_G}\right) = \boldsymbol{I}_d, \\
\boldsymbol{z}_i - \boldsymbol{z}_r &= \boldsymbol{R}^\top (\boldsymbol{x}_i - \boldsymbol{x}_r), \\
\| \boldsymbol{z}_i - \boldsymbol{z}_r \|_2^2 &= (\boldsymbol{x}_i - \boldsymbol{x}_r)^\top \boldsymbol{R}\boldsymbol{R}^\top (\boldsymbol{x}_i - \boldsymbol{x}_r), \\
&= \| \boldsymbol{x}_i - \boldsymbol{x}_r \|_2^2, \\
tr(\boldsymbol{R}^\top \boldsymbol{\Sigma} \boldsymbol{R}) &= tr(\boldsymbol{\Sigma}\boldsymbol{R}\boldsymbol{R}^\top), \\
&= tr(\boldsymbol{\Sigma}), \\
\boldsymbol{P} &= \boldsymbol{R}_q \boldsymbol{R}_q^\top, \\
\boldsymbol{P}^2 &= \boldsymbol{P}, \\
0 \preccurlyeq \boldsymbol{P} &\preccurlyeq \boldsymbol{I}, \\
\| \boldsymbol{R}_q^\top (\boldsymbol{x}_i - \boldsymbol{x}_r) \|_2 &\leq \| \boldsymbol{x}_i - \boldsymbol{x}_r \|_2.
\end{aligned} \tag{42}$$

A complete rotation changes coordinate orientation but not Euclidean geometry or total covariance trace. A truncated orthogonal projection can reduce a distance but cannot enlarge it. The classifier benefit comes from presenting rotated combinations to axis-aligned trees and from diversity across rotations, not from creating extra information. ▫

F. Algorithm 6: Calibrated Diversity Fusion

Algorithm 6 Calibrated Diversity Fusion and Selective Output (CDFS)

Input: inner out-of-fold NSVC and ERF probabilities; labels; utility coefficients; calibration family; review budget.

Output: fusion weights $\boldsymbol{\alpha}$; calibration parameters; attack posterior; accept/review decision.

Line	Operation
1	Align branch probabilities to the same class order and replace numerical zeros by ϵ.
2	Compute branch macro F1, multiclass log loss, calibration error, and useful disagreement.
3	Convert branch utilities to nonnegative softmax weights α_m.
4	Fuse out-of-fold probabilities and fit temperature or vector scaling on the fused logits.
5	Select (τ_p, τ_g) from inner data under the declared review-rate or selective-risk target.
6	Refit each branch on the complete outer training fold using selected settings.
7	At inference, obtain branch posteriors, compute the convex mixture, and apply the stored calibration.
8	Return the top class when confidence and probability-gap rules pass; return review otherwise.
9	Log branch disagreement, calibrated confidence, selected features, and local attribution for review.

Complexity: weight and calibration fitting are $O(nC)$ per iteration; inference adds $O(C)$ after the two branch predictions.

Protection: fusion and thresholds are based on out-of-fold training predictions; the outer test set is not employed in the selection of weights or coverage.

Ablation link: compare equal voting, F1-only weighting, uncalibrated utility weighting, calibrated fusion without review, and complete CDFS.

Proposition 6 (convex log-loss bound). For nonnegative weights that sum to one, the fused probability assigned to the true class has negative log loss no larger than the weighted mean branch log loss. The bound follows from convexity of $-log$.

Proof: For sample i with true class y_i, let $p_{m,i} = p_{i,y_i}^{(m)} > 0$. Then

$$\begin{aligned} \alpha_m &\geq 0, \\ \textstyle\sum_m \alpha_m &= 1, \\ p_i &= \textstyle\sum_m \alpha_m\, p_{m,i}, \\ \varphi(p) &= -logp, \\ \varphi''(p) &= 1/p^2 > 0, \\ \varphi\big(\textstyle\sum_m \alpha_m\, p_{m,i}\big) &\leq \textstyle\sum_m \alpha_m\, \varphi\big(p_{m,i}\big), \\ -logp_i &\leq \textstyle\sum_m \alpha_m \left[-logp_{m,i}\right], \\ \tfrac{1}{n}\textstyle\sum_i -logp_i &\leq \textstyle\sum_m \alpha_m \left[\tfrac{1}{n}\sum_i -logp_{m,i}\right]. \end{aligned} \tag{43}$$

The guarantee concerns log loss of the uncalibrated convex mixture and does not guarantee higher accuracy on every sample. Validation-calibrated weights can exploit complementary probabilities without exceeding the weighted average log loss. A later calibration map is retained only if it improves validation log loss and does not breach the macro-recall guardrail. ▫

G. End-to-End Leakage Proposition

Proposition 7 (outer-test noninterference). Algorithms 1-6 Assuming that the fit is performed for Each fitted option using only the outer training set and inner out-of-fold guessing, Then the outer test loss is a measurement of a predetermined learning algorithm rather than a function for which in the procedure of finding an optimum.

Proof: Let $\mathcal{A}$ denote the complete learning map. For outer fold k,

$$\begin{aligned} T_k &= T\big(\mathcal{D}_{k,tr}\big), \\ \mathcal{S}_k &= \mathcal{S}\left(T_k\big(\mathcal{D}_{k,tr}\big)\right), \\ \mathcal{B}_k &= \mathcal{B}\left(\mathcal{S}_k, T_k\big(\mathcal{D}_{k,tr}\big)\right), \\ \hat{\theta}_k &= InnerSelect(\mathcal{B}_k), \\ \big(F_{S,k}, F_{R,k}\big) &= Fit\big(\mathcal{B}_k, \hat{\theta}_k\big), \\ (\boldsymbol{\alpha}_k, \Gamma_k) &= Calibrate\big(OOF_{k,tr}\big), \\ F_k &= \mathcal{A}\big(\mathcal{D}_{k,tr}\big), \\ \hat{y}_i &= F_k(\boldsymbol{x}_i), \quad i \in \mathcal{I}_{k,te}, \\ \hat{L}_k &= \big|\mathcal{I}_{k,te}\big|^{-1} \textstyle\sum_{i\in\mathcal{I}_{k,te}} \ell\,(\hat{y}_i, y_i). \end{aligned} \tag{44}$$

All maps preceding $\hat{L}_k$ depend on $\mathcal{D}_{k,tr}$ and its inner partitions. The outer-test labels appear only in the final loss. This sequence prevents test-driven feature counts, sampling rates, model settings, fusion weights, or review thresholds. Dependence created by a flawed group definition remains possible, so group construction must be checked separately. ▫

VI. Experimental Design, Pilot Study, and Statistical Analysis

A. Benchmark Datasets

The baseline is constructed on the public NF-BoT-IoT and NF-ToN-IoT NetFlow datasets. Both are originated from packet captures and offer a unified flow format, however they present different priors and taxonomies for attacks. NF-BoT-IoT has five labels in the public version for the distribution analysis. NF-ToN-IoT consists of normal traffic plus nine attack labels. Raw IP addresses are not included as predictors. In case a certain version includes additional temporal or extended features, the version ID and schema hash are required with each run.

TABLE I Public Class Distributions Used for the Imbalance Analysis

Dataset	Class	Flows	Share (%)	Largest-to-class ratio
NF-BoT-IoT	Benign	13,859	2.310	33.96
	Reconnaissance	470,655	78.429	1.00
	DDoS	56,844	9.472	8.28

Dataset	Class	Flows	Share (%)	Largest-to-class ratio
	DoS	56,833	9.470	8.28
	Theft	1,909	0.318	246.55
NF-ToN-IoT	Benign	270,279	19.596	1.73
	Backdoor	17,247	1.250	27.17
	DoS	17,717	1.284	26.45
	DDoS	326,345	23.661	1.44
	Injection	468,539	33.970	1.00
	MITM	1,295	0.094	361.81
	Password	156,299	11.332	3.00
	Ransomware	142	0.010	3,299.57
	Scanning	21,467	1.557	21.83
	XSS	99,944	7.246	4.69

The numbers in Table I are obtained from the public dataset descriptions. They show why the task cannot be defined by accuracy. A classifier that simply predicts reconnaissance for all NF-BoT-IoT flows would be right 78.429% of the time, but it would not detect any theft, DoS, DDoS, or benign traffic. A classifier always predicting injection for all NF-ToN-IoT flow achieves 33.970% accuracy and zero recall for nine labels. The attack taxonomies also disallow a naive pooled label model without an explicit mapping between them.

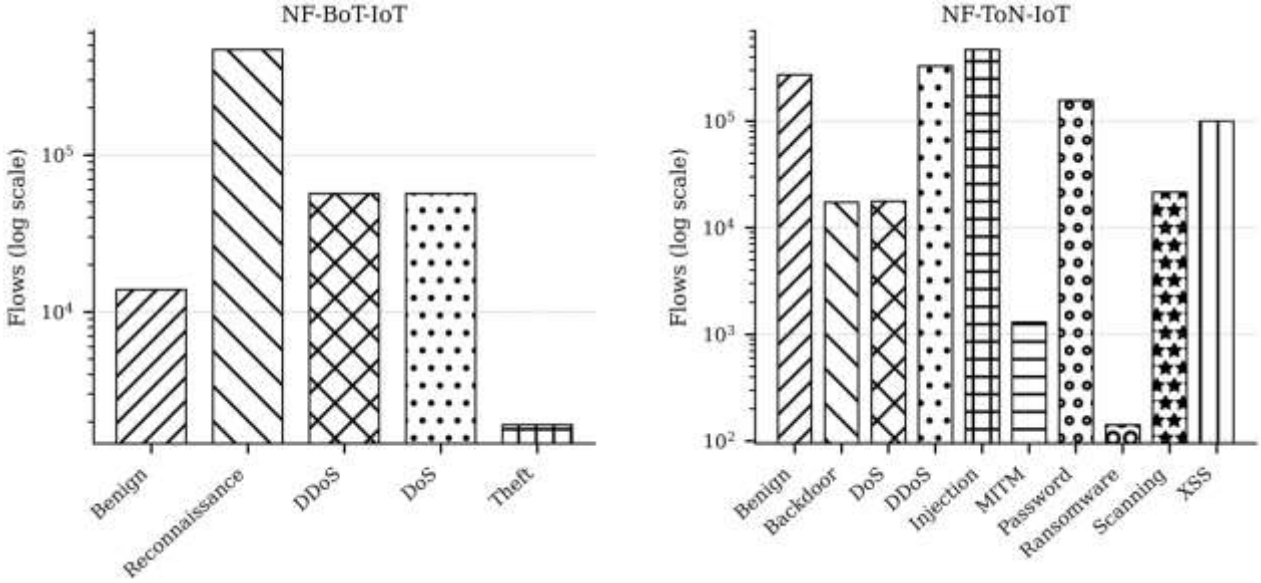


Fig. 2. The class counts for public in NF-BoT-IoT and NF-ToN-IoT are shown on a logarithmic axis. The y scale shows the theft and ransom ware tails that are almost invisible in the linear count plot.

B. Feature Schema and Leakage Controls

The default schema includes information such as the number of packets, number of bytes, duration, protocol, TCP state or flags, source and destination ports if allowed, directional rates, as well as NetFlow aggregate statistics available in the chosen public version. The execution log contains the ordered column names and data types. The classifier does not receive the raw source address, the raw destination address, the record sequence, the capture-file name, or the labels. Constant and quasi-constant variables are removed within a training fold. A correlation test detects deterministic transforms and duplicated columns.

Groups ought to represent attack scenario, capture file, time block, or endpoint when the metadata allow for such grouping. If one is restricted to row-level CSV data, a blocked temporal split is preferable to a random split. A

random stratified split may be stated as a secondary comparability assessment, although it is not a substitute for a group- or time-aware result. The model-selection loop never sees an outer-test row.

C. Baselines and Ablations

The comparison involves random forest, decision tree, XGBoost, unweighted SVC, class-weighted SVC, ordinary Rotation Forest, and the full MFRR-CEF. All baselines are subject to the same leakage-safe cleaning and outer folds. A second set of baselines implements each algorithm with its standard preprocessing only when this is specified as a relevant distinction. Hyperparameter budgets must be similar among methods.

The component ablation contains the follow six main rows: standard outlier protection, MFRR+ (no overlap), MFRR+ (no redundancy), standard SMOTE instead of BGMB, unweighted NSVC, standard Rotation Forest and majority-vote fusion. A cumulative ablation begins at LSRFP with an NSVC, then moves MFRR+, BGMB, S-ERF, calibration, and selective output. Each row contains the number of features, macro F1, macro recall, per-class recall, multiclass MCC, logloss, ECE, training time, flow throughput and memory.

D. Nested Validation Protocol

The recommended protocol is to use five outer grouped stratified folds with three inner grouped folds, and to repeat this over independent group assignments only if the group count allows. The fold manifest is set prior to training. Every outer fold computes a single held-out prediction for each flow. From inner folds: choose number of features and parameters by macro F1 subject to a min rare-class recall guardrail. Tie-breaking is done by smaller log loss, fewer features, and smaller inference cost, in that order.

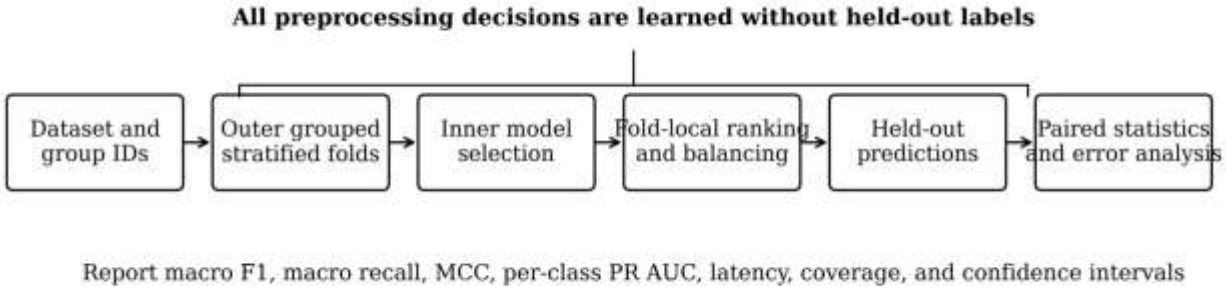


Fig. 3. Nested evaluation procedure Feature ranking, balancing, model selection, calibration and review-threshold selection are all performed on the training data. Predictions on held out data enable paired comparisons and error analysis.

E. Pilot Study

The pilot comprises two summaries which are tested over 699 samples. Pilot A states 674 correct and 25 misclassifications. B pilot states 687 correct and 12 misclassifications. The associated accuracy values are 0.964235 and 0.982833. The pilot files have generic class names instead of the official multiclass attack names, so the pilot can only supports a bi-modal or two-cluster diagnostic check and cannot derive NF-BoT-IoT or NF-ToN-IoT attack-wise accuracies.

For a correct count x out of n, the two-sided Wilson interval at nominal level $1-\alpha$ is

$$\frac{\hat{p}+z^2/(2n) \pm z\sqrt{\hat{p}(1-\hat{p})/n+z^2/(4n^2)}}{1+z^2/n}, \qquad (45)$$

where $\hat{p} = x/n$ and $z = 1.959964$ for 95%. Pilot A has interval [0.947736, 0.975659]; Pilot B has [0.970235, 0.990153]. The point difference is 0.018598, and the error count decreases from 25 to 12, a descriptive reduction of 52%. The same observations may have been used in both runs, but the discordant prediction counts are absent. A McNemar statistic cannot be computed from marginal correct totals.

The pilot also defines the gate for the full benchmark. A configuration progresses from diagnostic testing only when it produces finite probabilities for all 699 cases, reproduces its confusion totals from the saved predictions, and contains no duplicated test identifier. Metric consistency is checked by recomputing accuracy from correct counts, weighted recall from class supports, and MCC from the confusion matrix. The pilot file must preserve the fitted feature order so a changed column order cannot pass silently. A calibration check divides confidence into fixed bins and compares average confidence with observed correctness, subject to a minimum count per bin.

Clearing this gate doesn't mean we get performance on the public multiclass corpora. It establishes that the pipeline runs, prediction files are populated, the metrics are consistent internally, and the probability outputs can be used to do additional analysis later on. The next gate is a tiny multiclass run with all official labels and at least one complete capture group in each outer partition. The last gate is the full grouped nested benchmark. This tiered protocol stops a full experiment from shielding a defect in a schema, label-order, or probability-export that could have been detected on the pilot.

Pilot error review selects all 25 errors from Pilot A and all 12 errors from Pilot B, subject to matching row identifiers in the rerun. Each case receives true label, predicted label, confidence, probability gap, branch disagreement, nearest-class distances, and top feature contributions. Errors corrected by the later configuration are separated from regressions. This paired list supplies the n_{01} and n_{10} cells needed by McNemar's test and reveals whether the net gain comes from one class, one traffic group, or a broad reduction in error.

F. Statistical Testing Plan

For the complete benchmark, every model produces predictions on the same outer-test flows. The primary estimand is the paired difference in outer-fold macro F1. Secondary estimands include macro recall, MCC, log loss, ECE, per-class recall, and per-class precision-recall area. Dataset-level results remain separate; a combined estimate can use a prespecified random-effects or hierarchical model only when the execution includes more independent datasets.

The Friedman test is an omnibus test of ranks across folds or datasets for more than two classifiers, and is typically followed up with a multiplicity-controlled post-hoc procedure [52]. The fold dependence restricts a straightforward interpretation in repeated cross-validation, where training data is shared. Instead, paired permutation or bootstrap intervals over independent groups are preferred, provided sufficient feature groups are present. Common resampled tests have inflated type I error rates, as shown by Dietterich [53]. Nested selection is required because feature selection outside of cross-validation leads to optimistic error estimates [54].

For two classifiers on the same held-out flows, the 2×2 discordance table contains n_{01} cases wrong only for model 0 and n_{10} cases wrong only for model 1. McNemar's continuity-corrected statistic is

$$\chi^2 = \frac{(|n_{01}-n_{10}|-1)^2}{n_{01}+n_{10}}. \qquad (46)$$

THE TEST IS SECONDARY TO THE EFFECTSIZE AND CONFIDENCE INTERVALS. A macro-F1 interval is obtained by a group-stratified bootstrap, that draws complete sessions/capture blocks and not flows as the basic unit. Simultaneous intervals are presented for per-class recall differences or a false-discovery-rate adjustment is applied. Statistical significance does not align with operational value – the model must also satisfy false-positive, latency, and review-capacity constraints.

G. Reproducibility and Runtime Reporting

Every run has metadata, including dataset DOI or download URL, version, file hash, schema hash, group manifest, fold assignment, random seeds, software versions, processor, memory, training time, peak memory, and inference throughput. The train and test counts are given after every filter and before any artificial sampling. Per-class synthetic counts and reasons for rejection are reported. Feature ranks are saved per fold rather than just as one averaged ordering.

The run logs do not contain hardware metrics, so no latency or throughput value is reported. Full execution should evaluate batch-one latency at the target gateway or server, median and 95th-percentile flow processing time, model size, and flows per second. A learning system that improves macro F1 but that would exceed deployment budget is not operationally preferred.

G. Parameter Sensitivity and Minimum Evidence Rules

The full execution includes a prespecified sensitivity analysis for each parameter that controls model capacity or minority influence. MFRR+ is evaluated across feature counts $K \in \{5,10,15,20,d\}$, overlap estimators, redundancy limits, and rarity caps. The objective is not to select the curve point with the highest outer-test value. Inner folds select one configuration, and outer folds describe how that choice behaves. A sensitivity plot reports inner validation behavior and outer estimates with separate line styles so the selection process remains visible.

The density-overlap estimate requires adequate support. A fixed fine histogram can produce empty bins in the ransomware class. A broad kernel can hide a narrow region. The analysis varies grid resolution or bandwidth inside inner folds and records the effective number of occupied bins. When $n_c < n_0$, the quantile-overlap fallback is used and marked in the pair ledger. A pair score based on 142 ransomware flows is interpreted with greater uncertainty than one based on hundreds of thousands of injection flows.

BGMB sensitivity centers on the target parameter α, the local neighbor count k, and the class margin δ_c. Increasing α adds more candidates. Increasing δ_c rejects candidates near a competing class. A useful operating point has adequate acceptance, improved rare-class recall, and no material rise in benign false positives. The report includes attempts, acceptances, rejection causes, target shortfalls, and the distribution of accepted margins. It also verifies that no synthetic row appears in validation or test files.

Parameters of the NSVC are the kernel family, the penalty C, the kernel width, and the class-cost exponent. The size of the support set is reported as a measure of inference cost. A very small bandwidth radial kernel can memorize a testbed; a large bandwidth kernel merges small classes. ERF sensitivity contains tree count, group count, depth, leaf size, retained variance, and Fisher threshold. Performance should level off as the number of trees increase. A gain arising only from unrestricted depth also demands a duplicate and group-leakage check.

Fusion sensitivity examines macro-F1, log-loss, and diversity coefficients. A diversity reward is useful only when both components meet a minimum validation quality. The implementation sets a floor $F_m \geq F_{min}$ before rewarding disagreement. This prevents a weak random classifier from receiving weight merely for differing from a strong classifier. The calibration map is retained only when validation log loss improves and rare-class recall remains within its guardrail.

The minimum amount of empirical evidence for a performance claims consists of a dataset identifier and version, class map, feature list, excluded fields, group rule, fold manifest, random seed, model configuration, per-flow prediction table, and metric script. An accuracy value without a denominator or class map is reported as a descriptive only point estimate. A statement of significance should be based on paired predictions or independent units that fit the chosen test. A latency statement should specify the hardware, software version, batch size and measurement protocol. A cross-dataset statement has to share a fixed feature and label mapping prior to evaluation.

This makes the design of the model independent of the strength of the results. The comparative charts and pilot summaries specify the first recordings and enable numbers consistency verification. The full nested run is the entity that can demonstrate multiclass superiority, minority-class recall, calibration quality, and costs of operation.

VII. RESULTS AND INTERPRETATION

A. Class-Imbalance Analysis

The two datasets represent materially different learning problems. The NF-BoT-IoT dataset is heavily weighted towards reconnaissance (78.429% of flows). DDoS and DoS attacks each account for roughly 9.47% of flows, while benign traffic makes up 2.31% and theft 0.318%. The smallest class has enough samples to be fit with supervision, but the class has low influence on the outcome if the empirical risk is unweighted. A random row split of the dataset would result in many reconnaissance patterns being present in both training and test sets, potentially overstating performance if near-duplicate flows or attack windows are split between training and test.

NF-ToN-IoT has a wider taxonomy and longer minority tail. Injection accounts for 33.970%, DDoS 23.661%, benign 19.596%, password 11.332%, and XSS 7.246%. The other categories are much smaller: scanning 1.557%, DoS 1.284%, backdoor 1.250%, MITM 0.094% and ransomware 0.010%. The 142 ransomware flows cause a tight minimum-bottleneck of support. A high capacity classifier can achieve zero training error under a random split. A grouped split may put too few ransomware samples into a single training fold. The evaluation needs to report the number of trainable and testable samples for each of the outer folds, and should not hide a zero-support fold inside a pooled (macro) score.

No single oversampling ratio is suitable as can be seen from the log plot in Fig. 2. Multiplying ransomware by the injection count would also stretch it to more than 3,000 synthetic samples per original ransomware flow. Such an expansion would make the model rely on interpolation assumptions rather than on observations. BGMB employs the median class count as a bounded target and can also prematurely end at the retry limit. The class cost remains effective when synthesis fails to generate safe candidates.

Class entropy also differs between the datasets. For priors π_c, normalized entropy is

$$h = \frac{-\sum_{c=1}^{C} \pi_c log \pi_c}{log C}. \qquad (47)$$

A low h indicates concentration in a few labels; it does not reveal which rare pairs overlap. MFRR+ uses h as a global scaling term and retains pairwise weights for local discrimination. This prevents one entropy statistic from replacing class-specific evidence.

B. Comparative Classification Results

The comparative experiment shows the accuracy and precision of random forest, decision tree, XGBoost, and MFRR-CEF. The actual values that are plotted are shown in Table II. MFRR-CEF achieves 0.989 accuracy and 0.990 precision. The strongest baseline shown is the decision tree, which has 0.980 accuracy and 0.985 precision. Random forest has 0.970 and 0.974; XGBoost has 0.972 and 0.974.

TABLE II Comparative Classification Metrics

Method	Accuracy	Precision	Accuracy gap to MFRR-CEF	Precision gap to MFRR-CEF
Random forest	0.970	0.974	0.019	0.016
Decision tree	0.980	0.985	0.009	0.005
XGBoost	0.972	0.974	0.017	0.016
MFRR-CEF	**0.989**	**0.990**	—	—

The gain over the decision tree is 0.9 percentage points in accuracy and 0.5 percentage points in precision. The error rate associated with the accuracy is reduced from 0.020 to 0.011.The descriptive relative error reduction is

$$RER = \frac{(1-A_b)-(1-A_m)}{1-A_b} = \frac{A_m-A_b}{1-A_b}, \qquad (48)$$

which gives $(0.989 - 0.980)/(1 - 0.980) = 0.45$, or 45%, relative to the displayed decision-tree point estimate. This value is a transformation of aggregate accuracy, not a paired error analysis. Its operational meaning depends on the false-positive and false-negative classes that compose the remaining 1.1% error.

Precision increases from 0.985 to 0.990. If the precision plotted is the weighted or micro average, common classes dominate it. The plot does not specify its averaging rule. The full benchmark should give macro, weighted, micro, and class-wise precision separately. There are no recall or F1 scores in the comparison table, so one cannot infer from the table alone that detection of theft, ransomware, MITM, or backdoor got better.

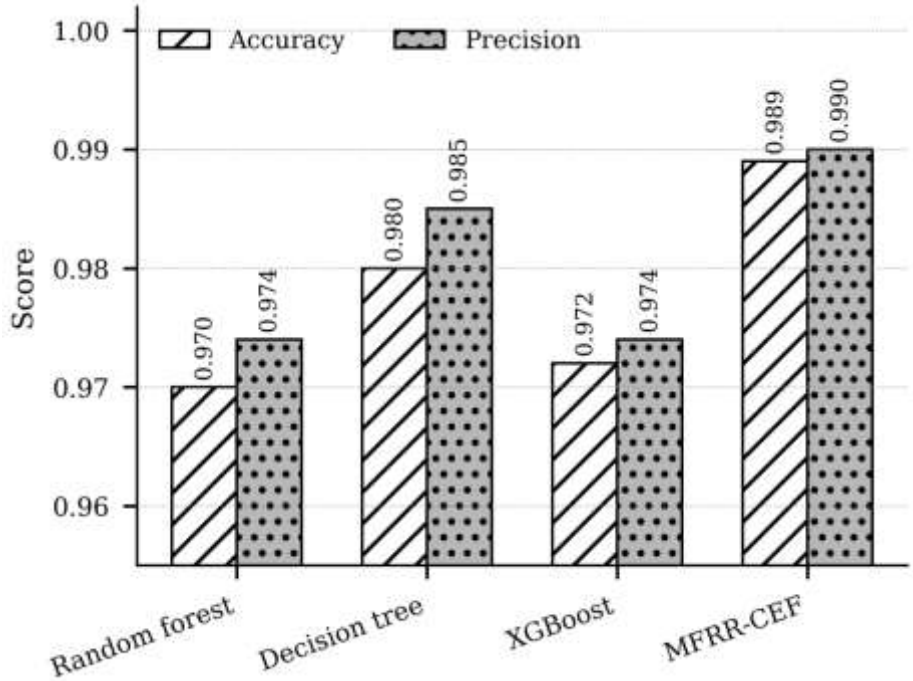


Fig. 4 Precision and accuracy in the comparative test. MFRR-CEF also has the highest point estimate for both measures. Error bars are not included as values at the fold level cannot be obtained.

The comparison provides a partial answer to RQ4: the full framework achieves better demonstrated accuracy and precision than the three baselines. Neither implies a statistically significant improvement, a cross-dataset gain, or a rare-class improvement. Such statements require common held-out predictions, dataset specific metrics, and paired protocol in Section VI-F.

C. Pilot Overall Performance

III presents the two 699-instance pilot summaries. Pilot A correctly classifies 674 instances and Pilot B 687. We obtain an improvement in accuracy of 1.8598 percentage points. Cohen's kappa goes up from 0.9211 to 0.9642, weighted F1 from 0.964 to 0.983 and MCC from 0.921 to 0.964. The mean absolute error drops from 0.0591 to 0.0301; the root mean square error falls from 0.1663 to 0.1115. The ROC area increases from 0.992 to 0.999 and the precision-recall area from 0.991 to 0.999.

TABLE III Pilot Results for 699 Instances

Metric	Pilot A	Pilot B	Descriptive change
Correct / total	674 / 699	687 / 699	+13 correct
Incorrect	25	12	−13 errors
Accuracy	0.964235	0.982833	+0.018598
95% Wilson interval	[0.947736, 0.975659]	[0.970235, 0.990153]	intervals overlap
Cohen’s kappa	0.9211	0.9642	+0.0431
Mean absolute error	0.0591	0.0301	−0.0290
Root mean square error	0.1663	0.1115	−0.0548
Weighted precision	0.964	0.983	+0.019
Weighted recall	0.964	0.983	+0.019
Weighted F1	0.964	0.983	+0.019
MCC	0.921	0.964	+0.043
ROC area	0.992	0.999	+0.007
PR area	0.991	0.999	+0.008

The Wilson intervals quantify the sampling uncertainty of each aggregate proportion under a binomial model. Their overlap is not a test of the difference. The accuracy comparison is therefore the function of how many cases are corrected and how many are missed by Pilot B, if the two are run on the same 699 flows. Ten fewer total errors could be obtained from 13 corrections and no regressions, 20 corrections and seven regressions, or other paired configurations. In each case, you get a different McNemar statistic. The actual paired predictions are required.

The decrease in squared error statistics implies that the probability or numeric output of Pilot B was closer to the encoded targets. What MAE and RMSE mean is dependent on how the evaluator treats class probability. They should not be interpreted as packet-count or latency errors. Kappa adjusts observed agreement by agreement expected predicted and true marginals. MCC summarizes confusion matrix. Both increase and this is in line with better balanced performance in the two-class diagnostic test.

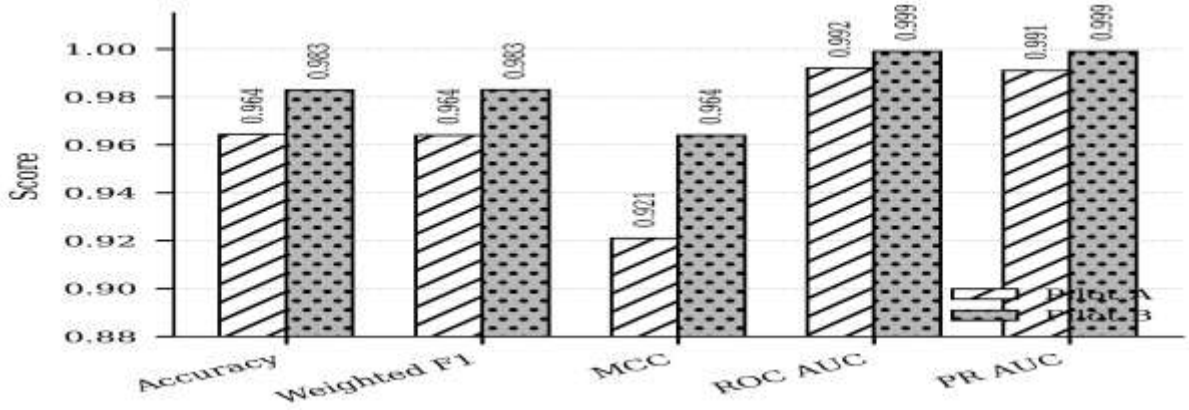


Fig. 5. Selected metrics from the two pilot summaries. Pilot B has higher accuracy, weighted F1, MCC, ROC area, and precision-recall area. The plot reports point values and does not substitute for paired inference.

D. Pilot Class-Level Results

The class names are not the same for all the pilot summaries. Pilot A with Class 1 and Class 2; Pilot B with Cluster 1 and Cluster 2. No mapping exists that links these tags to normal traffic or to a labeled attack. Table IV keeps the original generic labels. Pilot A obtains F1-scores of 0.973 and 0.948. Pilot B (0.978 and 0.986). The lower-performing class of Pilot A has precision and recall of 0.943 and 0.954. Lower class of Pilot B has recall 0.971, and both classes of Pilot B have precision > 0.98.

TABLE IV Pilot Class-Level Metrics

Run	Reported label	TP rate	FP rate	Precision	Recall	F1	MCC	ROC area	PR area
Pilot A	Class 1	0.969	0.046	0.976	0.969	0.973	0.921	0.992	0.996
Pilot A	Class 2	0.954	0.031	0.943	0.954	0.948	0.921	0.992	0.982
Pilot B	Cluster 1	0.971	0.010	0.986	0.971	0.978	0.964	0.999	0.998
Pilot B	Cluster 2	0.990	0.029	0.981	0.990	0.986	0.964	0.999	0.999

Pilot B reduces the false positive rate from 0.046 to 0.010 for the first reported class if Class 1 and Cluster 1 are the same target, but this is not explicitly stated. The only class-level claim that can be made with certainty is within each summary. Pilot B has two F1 scores of 0.978 or higher, while Pilot A has one class at 0.948. You will need actual labels and the associated support counts for a multiclass IoT claim.

E. Evidence Resolution and Claim Boundary

We have three types of result artifacts. The public dataset records have exact class counts. The graph of four methods has method-level point estimates of accuracy and precision. The pilot summaries have overall and two-class metrics for 699 instances. The resolutions cannot be combined to create a confusion matrix for the public multiclass datasets. Table V shows the allowed interpretation.

TABLE V Evidence Resolution and Supported Interpretation

Evidence	Available fields	Supported interpretation	Unsupported extension
Public dataset descriptions	class counts, labels, totals	imbalance ratios and evaluation design	achieved detector performance
Four-method chart	accuracy and precision point values	descriptive ordering of four methods	confidence interval, significance, rare-class recall
Pilot A summary	699 total, confusion-derived aggregate metrics	one-run binary/two-cluster diagnostic performance	named-attack performance
Pilot B summary	699 total, confusion-derived aggregate metrics	one-run binary/two-cluster diagnostic performance	paired superiority without predictions
Proposed nested execution	fold predictions, seeds, groups, timing	paired multiclass analysis after execution	current empirical claim

This separation eliminates two common reporting mistakes. The first is the association of pilot metrics to a named dataset without an experiment id. The second is to take a weighted metric as proof of minority-class performance. Each row of metrics for the final benchmark must be associated with a dataset version, a fold manifest, a label map, a feature schema, and a model configuration.

F. Answers to the Research Questions

Answer to RQ1: MFRR+ provides a mathematical resolution to common-class domination by aggregating finite pairwise contributions and balancing rarity weights. Its overlap term discounts a feature that has distinct means but wide common support. Its repetition denominator penalizes to much repeated reading of the same flow statistic. The current artifacts do not contain selected-features lists or fold ranks so empirical rank stability is to be determined. The overall evaluation will include selection frequency, median rank, pairwise contribution, and Jaccard stability over the outer folds.

Answer to RQ2: BGMB provides a label-preserving local test for accepting an interpolated sample. Proposition 3 shows that positive margin is robust to small perturbaitons. This addresses the geometric flaw of unconditional interpolation on the algorithmic level. Its empirical utility remains outstanding as the pilot findings do not single out the sampling design or the number of synthetics. The ablation will need to contrast vanilla SMOTE, bounded class weighting, and BGMB on the very same folds.
Answer to RQ3: NSVC and ERF are complementary in nature. NSVC adopts smooth kernel margins and explicit class costs. ERF is based on rotated attribute sets and piecewise tree partitions. Diversity fusion is beneficial only if the errors are different and both are well calibrated. The overall effect is consistent with a fusion gain but does not reveal any branch predictions. The execution plan will print disagreement, double-fault rate, Q-statistic, branch macro F1, and fused log loss.
Answer to RQ4: The displayed comparison favors MFRR-CEF, with 0.989 accuracy and 0.990 precision. The strongest displayed baseline has 0.980 accuracy and 0.985 precision. This is a positive descriptive result. It does not resolve statistical significance, dataset transfer, class-wise recall, calibration, or runtime. A submission-grade superiority claim depends on the nested paired experiment.
RQ5: The pilot demonstrates the feasibility of the classification pipeline on a 699-instance diagnostic subset. Pilot B makes 13 more correct decisions, an accuracy that is 1.8598 percentage points higher, a higher kappa and MCC, and a lower MAE and RMSE. The label map is not available and the predictions are only paired to the pilot data, which limits the conclusion to the pilot record. The solution is to rerun with the proper labels, a saved prediction table, group-aware folds, and all the relevant configuration metadata from Section IX.

VIII. DISCUSSION

A. Why Pairwise Ranking Matters

The public class counts show why a global relevance score can fail. In NF-ToN-IoT, a feature that separates injection from benign traffic can yield a large association statistic since those classes comprise more than half the corpus. Ransomware contributes 142 observations and has almost no effect on a frequency-weighted global score. MFRR+ gives every class pair a term and increases the relative contribution of pairs containing a small class, subject to a cap. The cap is needed: an inverse count of $1/142$ can make one unstable marginal density govern the rank.
The overlap term addresses a second defect. Consider two features with equal standardized mean separation. Feature j has narrow class distributions and little overlap; feature k has broad distributions and extensive overlap. A mean-only ratio treats them equally. MFRR+ multiplies by $1 - O_{j,ab}$ and assigns the larger pair contribution to j. If both features are strongly correlated, the greedy redundancy denominator tends to keep the higher raw score and discount the second. This combination supports a smaller model without assuming that one relevance statistic captures all behavior.
Rank interpretations should nevertheless be considered conditional. A large MFRR+ indicates that at least some class pairs in the training data are distinguished by the feature. This does not imply a causal link to an attack. A source port can be high while a testbed fixes a service; an address-based statistic can isolate one capture. A score should not be considered operationally stable without cross-time and cross-dataset validation. The contributions between pairs are stored in the reproducibility bundle, so an evaluator can see which class pair relationship gives rise to each rank.

B. Minority Support and Synthetic-Sample Risk

Theft and ransomware are two examples of different types of limitations. 1,909 NF-BoT-IoT flows are available for theft, which is enough for several folds if the attack flows are concentrated in a few classes. Ransomware has 142 NF-ToN-IoT flows, which may be from a few attack sessions. No sampling technique can generate independent attack sessions from 142 records. Interpolation can extend an observed local region, but the number of independent scenarios is still small.
BGMB views synthesis as regularization within supported regions. The class target is linked to the median count rather than the largest count. Candidate generation employs same-class neighborhoods. Structural constraints discard feasible combinations, and the competing-class margin discourages uncertain points. A small acceptance rate is an information: it means that this class lays close to other classes or it has no localsupport. The algorithm then depends on the bounded class cost and it reports the deficiency instead of generating unchecked samples.

Categorical flow attributes require special handling. For duration or a scaled byte count, linear interpolation is meaningful, but not for protocol, TCP state, or a binary flag. BGMB samples a categorical value from the local combinations observed, or from a rule-constrained conditional distribution. Derived numeric fields must be internally consistent. A candidate with zero duration and a positive packets-per-second value, for example, contradicts its defining relation. The implementation checks these relations for consistency after synthesis.

C. Complementarity of NSVC and ERF

NSVC and ERF react in different ways to feature geometry. A radial-basis SVC can form smooth nonlinear boundaries enclosing regions of attack. Its support-vector number may become large in overlapping data, thereby increasing inference cost. The class costs shift the distribution of penalties but do not introduce any new information about an unsupported class. Calibration turns pair margins into probabilities on which the fusion layer operates.

ERF rotates correlated flow features prior to tree construction. A regular decision tree performs axis-aligned splits, it may take many levels to approximate an oblique relation between duration, packets and bytes. An orthonormal rotation can be used to align that relation with a tree axis. Various bootstraps andgroups yield various rotated views. Class-supervised grouping and Fisher-guided retention concentrate rotations on class-separating variance, but they can overfit a small class if fitted outside an inner folded. The two level procedure helps to reduce that risk. Fusion gains come from corrected errors of disagreement. If both agents are making the same errors, a weighted average is of little use. When one branch is correct but overconfident and the other one is less accurate but well calibrated, a F1-only weighting scheme can degrade the log loss. CDFS contains validation log loss as well as good disagreement. Nonnegative weights maintain the interpretability of the mixture and endow the convex log-loss property from Proposition 6. The mixture also require a held-out check for macro recall and calibration.

D. Interpretation of Accuracy and Precision

0.989 in Table II is a high accuracy, but class imbalance changes its interpretation. In relation to the average, one percentage point means 6,001 errors on the basis of all 600,100 flows of NF-BoT-IoT and 13,793 errors on the basis of all 1,379,274 flows of N f-T o N- I o T. The comparative graph does not indicate the sample size evaluated or the dataset version, thus these products are illustrations rather than error counts for the experiment. Operational reports should report counts, rates, and class identities.

Precision of 0.990 also needs to be averaged definition-wise. Micro precision in single-label multiclass is the same as accuracy. Weighted precision favors common labels. Macro precision is the average precision per class, however it can fluctuate a lot if a rare class has a few positive predictions. Precision and recall per class are needed. The precision-recall area is good for rare one-versus-rest evaluations but has to be interpreted with class prevalence.

Pilot summaries of MCC and kappa support the two-class diagnostic interpretation. Pilot B has MCC 0.964 vs Pilot A 0.921. The pilot class F1 values also improve. However, the generic labels do not allow a conclusion specific to the attack. Labels are kept the same to avoid mapping to named attacks without evidence. Next run should export the confusion matrix with official class order.

E. Error Taxonomy

An intrusion detector can fail due to label confusion, distribution shift, representation failure, calibration failure, or throughput failure. Label confusion is where behaviors have similar flow statistics, e.g., DDoS and DoS or scanning and reconnaissance. Representation failure is where the selected NetFlow schema lacks the time-series or payload evidence required to distinguish a class. Distribution shift is where the devices, protocols, addresses, rates, or benign workloads shift. Calibration failure is where the reported confidence does not correspond to accuracy. Throughput failure is where the processing time is too long such that the decision is made after the flow has already done damage.

Error analysis should stratify held-out mistakes by true class, predicted class, confidence, group, time, protocol, and selected-feature region. For each confusion pair (a, b), define normalized error mass

$$E_{ab} = \frac{N_{ab}}{\sum_{r \neq a} N_{ar}}, \qquad a \neq b, \tag{49}$$

where N_{ab} is the confusion count. High E_{ab} identifies the dominant destination of class-a errors. A feature-overlap report then compares $O_{j,ab}$ for selected features. This links a classifier error to the ranking evidence rather than reporting a confusion matrix alone.

False positives increase analyst workload and automated response. A benign flow that is mislabeled as DDoS can cause unnecessary blocking. False negatives impact security exposure. A ransomware or exfiltration flow that is mislabeled as benign can be high impact. The deployment threshold can be used to apply different review rules to

classes that have higher impact. These costs must be explicitly defined before performing a test, not inferred from the test result.

F. Calibration and Review Queue

A calibrated posterior enables a reviewer to order the flows in terms of estimated risk, although under distribution shift one scalar confidence is not enough. The output log must contain top class, top probability, top-two gap, branch probabilities, disagreement, selected-feature attribution and an out-of-distribution score. When confidence is low, branches disagree, or the selected feature vector is far away from training support, a flow may be submitted for review.

Beside selective risk, coverage has to be reported. A rule can get a very low accepted error rate by examining almost all flows. This tradeoff is illustrated by the risk-coveragecurve. Per-class coverage is required as well. A threshold that accepts all common reconnaissance flows and eliminates all theft flows might show high overall selective risk but very poor security utility.

Temperature scaling is easy and preserves the number of classes, but one parameter may not be enough for a rare class. Vector scaling or classwise calibration is possible, for a given validation support. Ransomware may have a small number of validation samples which is too low to allow for an independent map. Then the model should keep a conservative global map and report wide uncertainty rather than fitting an unstable class-specific calibrator.

G. Cross-Dataset Generalization

Hyperplane-based methods NF-BoT-IoT and NF-ToN-IoT share a NetFlow representation, but differ in testbed, attacks, benign workload and priors. One cannot simply assume that a model trained on one set will perform well on the other. Three tests are recommended. The first uses grouped folds to train and test within each dataset. The second is trained on one dataset and tested only for common labels and features on another. The third is trained on the union with a dataset indicator marginalized out of prediction and tested on a held-out dataset.

There should be explicit label mapping. Reconnaissance is a broad semantic cover term that can be equal with scanning, but the process of data collection in these two activities may vary. Theft is not in the NF-ToN-IoT taxonomy; ransomware, MITM, backdoor, injection, password and XSS are not in the NF-BoT-IoT. The punctured intersection taxonomy can be used for cross-dataset multiclass experiments. It cannot quietly take every class that doesn't match, map it to attack, and do multiclass transfer.

Feature stability can be measured across folds and datasets. For selected sets $\mathcal{S}_r$ and $\mathcal{S}_s$, Jaccard stability is

$$J_{rs} = \frac{|\mathcal{S}_r \cap \mathcal{S}_s|}{|\mathcal{S}_r \cup \mathcal{S}_s|}. \qquad (50)$$

Rank correlation is useful when sets have the same members but in a different order. Low stability can indicate real differences in the underlying distributions, lack of support from the minority, or that the score is too sensitive. The pairwise score ledger assists in identifying these causes.

H. Computational and Deployment Considerations

The only feature-engineering cost is the overlap in densities MFRR+. A pairwise direct kernel-density calculation over millions of flows is not needed. The implementation may use stratified quantile sketches or fixed histograms for each class and each feature. Streaming counts saves memory, and candidate screening will also remove attributes that are constant or essentially equal across the population before correlation calculation. Approximate Spearman correlations can be computed on a fixed training subsample, and exact tests can be run among selected candidates.

NSVC kernel learning can be costly with millions of flows. The framework provides a linear or an approximate kernel branch for large sets of users, with the same class-cost and calibration logic. ERF training parallelizes per tree . Inference can be done as a streaming transformation by branch prediction. The final model should report throughput at batch size one and at the production batch size.

Model updates are done with a versioned pipeline. On every new traffic window, the pipeline checks for schema drift, missing categories, prior drift, feature drift, and error drift (labels are now available). Models are not retrained on live traffic without review. Training and evaluation hashes are retained and linked to each reported value. If calibration or false positive rates degrade, the previous model is still available to roll back to.

X. CONCLUSION

MFRR-CEF integrates leak-free flow preprocessing, overlapped and entropy-aware feature ranking, guarded class balancing, class-dependent support vector machine margins, supervised Rotation Forest projections and calibrated diversity fusion in a multi-class IoT intrusion detection system. It considers main building blocks of NF-BoT-IoT and

NF-ToN-IoT such as the unequal class support, large pairwise overlap, correlated NetFlow features, and different attack taxonomies.

The mathematical analysis establishes bounded feature scores, local safety of accepted synthetic samples, bounded minority margin influence, distance preservation under complete rotations, a convex log-loss bound for fusion, and outer-test isolation under nested evaluation. The comparison places MFRR-CEF above random forest, decision tree, and XGBoost, with accuracy of 0.989 and precision of 0.990. The pilot record shows an increase from 674 to 687 correct decisions over 699 instances and an accuracy increase from 0.9642 to 0.9828. These values support feasibility and descriptive improvement. They do not replace class-labeled, fold-level, paired validation.

The full evaluation requires grouped nested folds, official multiclass labels, saved per-flow probabilities, class-wise precision-recall analysis, calibration, ablation, runtime measurement, and cross-dataset testing. The code and data tables make the reported figures traceable and define the missing execution artifacts. This evidence boundary permits the current results to be read accurately and gives a complete path for final benchmark validation.

REFERENCES

[1] U. K. Lilhore, S. Dalal, and S. Simaiya, "A cognitive security framework for detecting intrusions in IoT and 5G utilizing deep learning," Computers & Security, vol. 136, Art. no. 103560, Jan. 2024, doi: 10.1016/j.cose.2023.103560.

[2] C. Alex, G. Creado, W. Almobaideen, O. A. Alghanam, and M. Saadeh, "A comprehensive survey for IoT security datasets taxonomy, classification and machine learning mechanisms," Computers & Security, vol. 132, Art. no. 103283, Sep. 2023, doi: 10.1016/j.cose.2023.103283.

[3] G. A. Mukhaini, M. Anbar, S. Manickam, T. A. Al-Amiedy, and A. A. Momani, "A systematic literature review of recent lightweight detection approaches leveraging machine and deep learning mechanisms in Internet of Things networks," J. King Saud Univ.—Comput. Inf. Sci., vol. 36, no. 1, Art. no. 101866, Jan. 2024, doi: 10.1016/j.jksuci.2023.101866.

[4] A. Nazir et al., "Advancing IoT security: A systematic review of machine learning approaches for the detection of IoT botnets," J. King Saud Univ.—Comput. Inf. Sci., vol. 35, no. 10, Art. no. 101820, Dec. 2023, doi: 10.1016/j.jksuci.2023.101820.

[5] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," Computers & Security, vol. 86, pp. 147–167, Sep. 2019, doi: 10.1016/j.cose.2019.06.005.

[6] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in Proc. IEEE Symp. Security and Privacy, 2010, pp. 305–316, doi: 10.1109/SP.2010.25.

[7] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," Future Gener. Comput. Syst., vol. 100, pp. 779–796, Nov. 2019, doi: 10.1016/j.future.2019.05.041.

[8] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets," Sustainable Cities Soc., vol. 72, Art. no. 102994, Sep. 2021, doi: 10.1016/j.scs.2021.102994.

[9] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems," IEEE Access, vol. 8, pp. 165130–165150, 2020, doi: 10.1109/ACCESS.2020.3022862.

[10] T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. H. den Hartog, "ToN IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion datasets," IEEE Internet Things J., vol. 9, no. 1, pp. 485–496, Jan. 2022, doi: 10.1109/JIOT.2021.3085194.

[11] M. Sarhan, S. Layeghy, N. Moustafa, and M. Portmann, "NetFlow datasets for machine learning-based network intrusion detection systems," in Big Data Technologies and Applications, Cham, Switzerland: Springer, 2021, pp. 117–135, doi: 10.1007/978-3-030-72802-1_9.

[12] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a standard feature set for network intrusion detection system datasets," Mobile Netw. Appl., vol. 27, pp. 357–370, 2022, doi: 10.1007/s11036-021-01843-0.

[13] H. He and E. A. Garcia, "Learning from imbalanced data," IEEE Trans. Knowl. Data Eng., vol. 21, no. 9, pp. 1263–1284, Sep. 2009, doi: 10.1109/TKDE.2008.239.

[14] P. Branco, L. Torgo, and R. P. Ribeiro, "A survey of predictive modeling on imbalanced domains," ACM Comput. Surv., vol. 49, no. 2, Art. no. 31, Aug. 2016, doi: 10.1145/2907070.

[15] B. Krawczyk, “Learning from imbalanced data: Open challenges and future directions,” Prog. Artif. Intell., vol. 5, pp. 221–232, 2016, doi: 10.1007/s13748-016-0094-0.
[16] N. Japkowicz and S. Stephen, “The class imbalance problem: A systematic study,” Intell. Data Anal., vol. 6, no. 5, pp. 429–449, 2002, doi: 10.3233/IDA-2002-6504.
[17] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.
[18] H. He, Y. Bai, E. A. Garcia, and S. Li, “ADASYN: Adaptive synthetic sampling approach for imbalanced learning,” in Proc. IEEE Int. Joint Conf. Neural Networks, 2008, pp. 1322–1328, doi: 10.1109/IJCNN.2008.4633969.
[19] C. Seiffert, T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, “RUSBoost: A hybrid approach to alleviating class imbalance,” IEEE Trans. Syst., Man, Cybern. A, vol. 40, no. 1, pp. 185–197, Jan. 2010, doi: 10.1109/TSMCA.2009.2029559.
[20] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera, “A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches,” IEEE Trans. Syst., Man, Cybern. C, vol. 42, no. 4, pp. 463–484, Jul. 2012, doi: 10.1109/TSMCC.2011.2161285.
[21] L. Breiman, “Random forests,” Mach. Learn., vol. 45, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
[22] J. J. Rodríguez, L. I. Kuncheva, and C. J. Alonso, “Rotation Forest: A new classifier ensemble method,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 10, pp. 1619–1630, Oct. 2006, doi: 10.1109/TPAMI.2006.211.
[23] C. Cortes and V. Vapnik, “Support-vector networks,” Mach. Learn., vol. 20, pp. 273–297, 1995, doi: 10.1007/BF00994018.
[24] M. Robnik-Šikonja and I. Kononenko, “Theoretical and empirical analysis of ReliefF and RReliefF,” Mach. Learn., vol. 53, pp. 23–69, 2003, doi: 10.1023/A:1025667309714.
[25] H. Peng, F. Long, and C. Ding, “Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 27, no. 8, pp. 1226–1238, Aug. 2005, doi: 10.1109/TPAMI.2005.159.
[26] I. Guyon and A. Elisseeff, “An introduction to variable and feature selection,” J. Mach. Learn. Res., vol. 3, pp. 1157–1182, Mar. 2003.
[27] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation Forest,” in Proc. 8th IEEE Int. Conf. Data Mining, 2008, pp. 413–422, doi: 10.1109/ICDM.2008.17.
[28] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” ACM Comput. Surv., vol. 41, no. 3, Art. no. 15, Jul. 2009, doi: 10.1145/1541880.1541882.
[29] O. Abu Alghanam, W. Almobaideen, M. Saadeh, and O. Adwan, “An improved PIO feature selection algorithm for IoT network intrusion detection system based on ensemble learning,” Expert Syst. Appl., vol. 213, Art. no. 118745, Mar. 2023, doi: 10.1016/j.eswa.2022.118745.
[30] N. Prazeres, R. L. de C. Costa, L. Santos, and C. Rabadão, “Engineering the application of machine learning in an IDS based on IoT traffic flow,” Intell. Syst. Appl., vol. 17, Art. no. 200189, Feb. 2023, doi: 10.1016/j.iswa.2023.200189.
[31] T. Saba, A. Rehman, T. Sadad, H. Kolivand, and S. A. Bahaj, “Anomaly-based intrusion detection system for IoT networks through deep learning model,” Comput. Electr. Eng., vol. 99, Art. no. 107810, Apr. 2022, doi: 10.1016/j.compeleceng.2022.107810.
[32] B. Sharma, L. Sharma, C. Lal, and S. Roy, “Anomaly based network intrusion detection for IoT attacks using deep learning technique,” Comput. Electr. Eng., vol. 107, Art. no. 108626, Apr. 2023, doi: 10.1016/j.compeleceng.2023.108626.
[33] R. Lazzarini, H. Tianfield, and V. Charissis, “A stacking ensemble of deep learning models for IoT intrusion detection,” Knowl.-Based Syst., vol. 279, Art. no. 110941, Nov. 2023, doi: 10.1016/j.knosys.2023.110941.
[34] Y. Cao, Z. Wang, H. Ding, J. Zhang, and B. Li, “An intrusion detection system based on stacked ensemble learning for IoT network,” Comput. Electr. Eng., vol. 110, Art. no. 108836, Sep. 2023, doi: 10.1016/j.compeleceng.2023.108836.
[35] R. K. Muna, M. I. Hossain, M. G. R. Alam, M. M. Hassan, M. Ianni, and G. Fortino, “Demystifying machine learning models of massive IoT attack detection with explainable AI for sustainable and secure future smart cities,” Internet Things, vol. 24, Art. no. 100919, Dec. 2023, doi: 10.1016/j.iot.2023.100919.
[36] N. F. Syed, M. Ge, and Z. Baig, “Fog-cloud based intrusion detection system using recurrent neural networks and feature selection for IoT networks,” Comput. Netw., vol. 225, Art. no. 109662, Apr. 2023, doi: 10.1016/j.comnet.2023.109662.

[37] O. Kornyo et al., “Botnet attacks classification in AMI networks with recursive feature elimination and machine learning algorithms,” Computers & Security, vol. 135, Art. no. 103456, Dec. 2023, doi: 10.1016/j.cose.2023.103456.
[38] M. M. Alani, “BotStop: Packet-based efficient and explainable IoT botnet detection using machine learning,” Comput. Commun., vol. 193, pp. 53–62, Sep. 2022, doi: 10.1016/j.comcom.2022.06.039.
[39] M. Sarhan, S. Layeghy, N. Moustafa, M. Gallagher, and M. Portmann, “Feature extraction for machine learning-based intrusion detection in IoT networks,” Digit. Commun. Netw., vol. 10, no. 1, pp. 205–216, Feb. 2024, doi: 10.1016/j.dcan.2022.08.012.
[40] O. Habibi, M. Chemmakha, and M. Lazaar, “Imbalanced tabular data modelization using CTGAN and machine learning to improve IoT botnet attacks detection,” Eng. Appl. Artif. Intell., vol. 118, Art. no. 105669, Feb. 2023, doi: 10.1016/j.engappai.2022.105669.
[41] S. Li, Y. Cao, S. Liu, Y. Lai, Y. Zhu, and N. Ahmad, “HDA-IDS: A hybrid DoS attacks intrusion detection system for IoT by using semi-supervised CL-GAN,” Expert Syst. Appl., vol. 238, Art. no. 122198, Mar. 2024, doi: 10.1016/j.eswa.2023.122198.
[42] E. Ntizikira, L. Wang, J. Chen, and K. Saleem, “Honey-block: Edge assisted ensemble learning model for intrusion detection and prevention using defense mechanism in IoT,” Comput. Commun., vol. 214, pp. 1–17, Jan. 2024, doi: 10.1016/j.comcom.2023.11.023.
[43] A. M. Araujo, A. Bergamini de Neira, and M. Nogueira, “Autonomous machine learning for early bot detection in the Internet of Things,” Digit. Commun. Netw., vol. 9, no. 6, pp. 1301–1309, Dec. 2023, doi: 10.1016/j.dcan.2022.05.011.
[44] J. Ashraf et al., “IoTBoT-IDS: A statistical learning-enabled botnet detection framework for protecting networks of smart cities,” Sustainable Cities Soc., vol. 72, Art. no. 103041, Sep. 2021, doi: 10.1016/j.scs.2021.103041.
[45] N. Moustafa, M. Keshk, K.-K. R. Choo, T. Lynar, S. Camtepe, and M. Whitty, “DAD: A distributed anomaly detection system using ensemble one-class statistical learning in edge networks,” Future Gener. Comput. Syst., vol. 118, pp. 240–251, May 2021, doi: 10.1016/j.future.2021.01.011.
[46] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in Proc. 34th Int. Conf. Machine Learning, 2017, pp. 1321–1330.
[47] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in Advances in Neural Information Processing Systems 30, 2017, pp. 4765–4774.
[48] T.-F. Wu, C.-J. Lin, and R. C. Weng, “Probability estimates for multi-class classification by pairwise coupling,” J. Mach. Learn. Res., vol. 5, pp. 975–1005, Aug. 2004.
[49] J. C. Platt, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” in Advances in Large Margin Classifiers, A. J. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. Cambridge, MA, USA: MIT Press, 1999, pp. 61–74.
[50] T. Saito and M. Rehmsmeier, “The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets,” PLoS ONE, vol. 10, no. 3, Art. no. e0118432, Mar. 2015, doi: 10.1371/journal.pone.0118432.
[51] J. Gorodkin, “Comparing two K-category assignments by a K-category correlation coefficient,” Comput. Biol. Chem., vol. 28, nos. 5–6, pp. 367–374, Dec. 2004, doi: 10.1016/j.compbiolchem.2004.09.006.
[52] J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” J. Mach. Learn. Res., vol. 7, pp. 1–30, Jan. 2006.
[53] T. G. Dietterich, “Approximate statistical tests for comparing supervised classification learning algorithms,” Neural Comput., vol. 10, no. 7, pp. 1895–1923, Oct. 1998, doi: 10.1162/089976698300017197.
[54] S. Varma and R. Simon, “Bias in error estimation when using cross-validation for model selection,” BMC Bioinformatics, vol. 7, Art. no. 91, Feb. 2006, doi: 10.1186/1471-2105-7-91.